

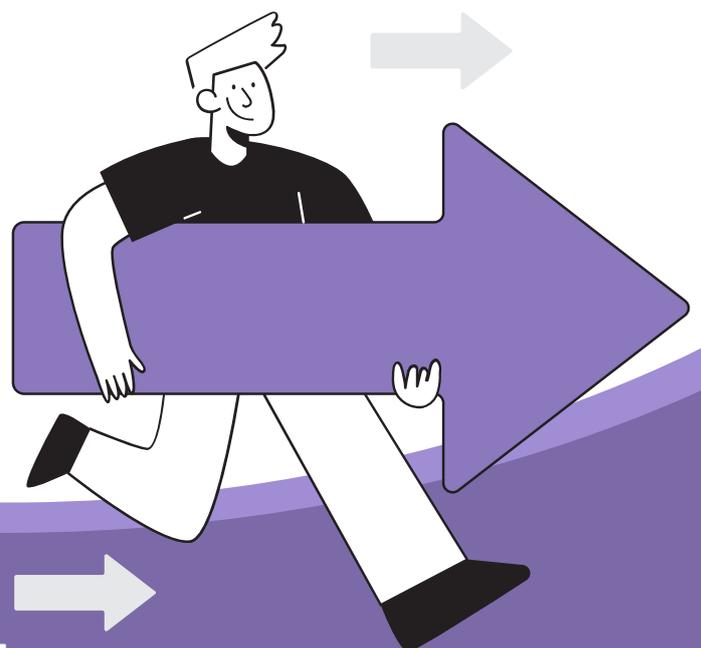
Programme Formation

Kotlin - Avancé



Kotlin est un langage de programmation moderne développé par JetBrains. Ce langage statiquement typé peut être exécuté sur la JVM et est complètement interopérable avec Java. Il offre une syntaxe concise et des fonctionnalités modernes qui facilitent le développement d'applications, comme une meilleure gestion de la nullité, des capacités d'extensions et une API très riche sur les collections...

Cette formation a pour objectif de transmettre toutes les connaissances fondamentales de Kotlin afin de produire un code clair et expressif. Les développeurs pourront ainsi produire des applications efficaces et maintenables en s'appuyant sur la puissance du SDK de Kotlin, mais également de Java . La formation s'appuie sur des éléments théoriques mais également des démonstrations et des travaux pratiques pour assurer une maîtrise complète des concepts fondamentaux de ce langage.



Sommaire

01 Objet, nature, effectif de la formation

- 1.1 - Intitulé de la formation
- 1.2 - Objectifs pédagogiques
- 1.3 - Compétences visées
- 1.4 - Public concerné
- 1.5 - Prérequis
- 1.6 - Modalités d'évaluation et de suivi
- 1.7 - Modalités pédagogiques et techniques
- 1.8 - Référent pédagogique

02 Programme

03 Lieu, participants, prix

- 3.1 - Lieu de la formation
- 3.2 - Participants
- 3.3 - Durée et prix de la formation



01

Objet, nature, effectif de la formation

◆ 1.1 - Intitulé de la formation

Kotlin - Avancé

◆ 1.2 - Objectifs pédagogiques

- Maîtriser les idioms Kotlin
- Savoir optimiser une fonction Kotlin en utilisant la "tail recursion" ou "inline"
- Savoir utiliser les interfaces fonctionnelles
- Maîtriser les fonctionnalités avancées de Kotlin pour exprimer un DSL
- Comprendre l'interopérabilité avec Java
- Maîtriser l'usage des coroutines
- Savoir faire communiquer des coroutines
- Savoir traiter des flux de données asynchrones
- Maîtriser l'écriture de code générique
- Comprendre la délégation pour favoriser la composition
- Savoir utiliser la méta programmation

◆ 1.3 - Compétences visées

- Utiliser un singleton
- Utiliser les expressions
- Utiliser la null-safety
- Utiliser les Data class
- Interpoler des chaînes de caractère
- Utiliser une scope fonction
- Utiliser l'api collections
- Générer une séquence
- Optimiser une fonction récursive
- Optimiser l'usage d'une fonction en utilisant inline
- Déclarer et utiliser une interface fonctionnelle comme une lambda
- Faciliter la lecture des fonctions monoparamètre avec infix
- Exprimer des types avec typealias
- Étendre des fonctions ou des propriétés
- Exprimer simplement des builders avec les lambda et les receveurs
- Gérer correctement la nullité issue de Java
- Comprendre le mapping des concepts Java
- Décrire la compatibilité avec les annotations
- Utiliser la réflexion Java
- Travailler avec du code non-bloquant
- Comprendre la concurrence structurée
- Construire une coroutine et exécuter fonction "suspend"
- Définir le contexte et le dispatcher d'une coroutine
- Contrôler la fin d'une coroutine
- Utiliser les channels pour faire communiquer les coroutines
- Créer un flux de données asynchrones
- Appliquer des opérations de transformations sur les flux
- Composer des flux asynchrones

◆ 1.3 - Compétences visées

- Déclarer des génériques dans une classe ou une méthode
- Déclarer correctement les limites des types génériques
- Réifier un type générique pour éviter l'effacement de type
- Comprendre quand utiliser la composition
- Déléguer une interface ou une propriété
- Utiliser la réflexion en Kotlin pour définir des traitements spécifiques
- Utiliser les annotations pour générer du code
- Utiliser les plugins processeur de symboles pour générer du code
- Utiliser les plugins de compilateur pour générer du code

◆ 1.4 - Public concerné

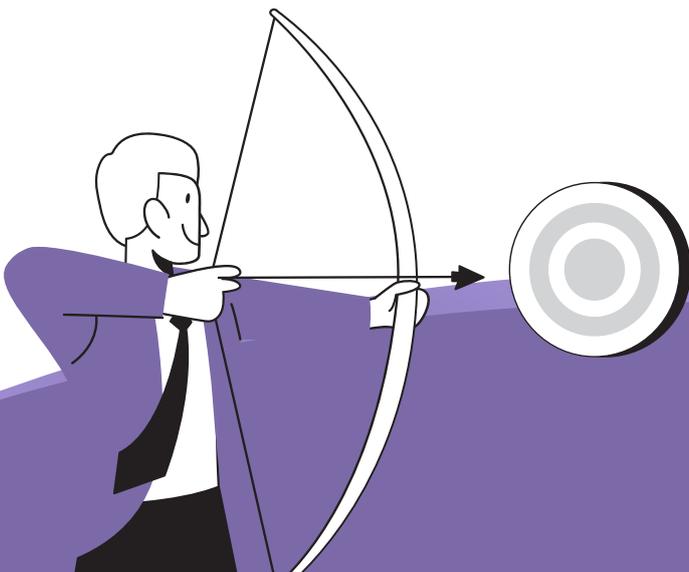
Cette formation s'adresse à des développeurs Kotlin qui souhaitent parfaire leur maîtrise du langage en produisant du code plus robuste et maintenable.

◆ 1.5 - Prérequis

- Maîtriser les bases de la programmation objet
- Maîtriser les bases du langage Kotlin

◆ 1.6 - Modalités d'évaluation et de suivi

- Feuilles de présence signées des participants et du formateur par demi-journée ;
- Attestation de fin de formation mentionnant les objectifs, la nature et la durée de l'action et les résultats de l'évaluation des acquis de la formation.



◆ 1.7 - Modalités pédagogiques et techniques

MODALITÉS PÉDAGOGIQUES :

- Évaluation des besoins et du profil du participant
- Apport théorique et méthodologique : séquences pédagogiques regroupées en différents modules
- Contenus des programmes adaptés en fonction des besoins identifiés pendant la formation
- Questionnaires, exercices et étude de cas
- Réflexion et échanges sur cas pratiques
- Retours d'expériences

ÉLÉMENTS MATÉRIELS :

- Mise à disposition de tout le matériel informatique et pédagogique nécessaire (hors PC)
- Support de cours au format numérique projeté sur écran et transmis au participant par mail à la fin de la formation

◆ 1.8 - Référent pédagogique

Chaque formation est sous la responsabilité du directeur pédagogique de l'organisme de formation ; le bon déroulement est assuré par le formateur désigné par l'organisme de formation.



02

Programme

1. Introduction

2. Retours sur les idioms Kotlin

- Class
- Singleton
- Expressions
- Null Safety
- Paramètre nommées, par défaut
- Data class
- Interpolation et chaînes de caractère multiples
- Lambda
- Collections Immutables et Mutables
- Api Collections
- Séquence ou Iterable
- Scope Function
- Range
- Générer une séquence

3. Plus loin avec les fonctions et les lambdas

- tail recursion
- Les fonctions locales
- inline
- Interface fonctionnelle

4. Créer un DSL

- “infix” pour simplifier la lecture
- Exprimer des type avec “type alias”
- Extension des fonctions et des propriétés
- Surcharge d’opérateur
- invoke avec un companion object
- La “Lambda” comme dernier paramètre
- Le receveur de “Lambda”

5. Interopérabilité avec Java

- Gestion de “null”
- La mapping des concepts Java
- Les annotations de compatibilité
- La réflexion Java

6. Introduction au coroutines

- Une coroutine
- Travailler avec du non bloquant
- Concurrence structurée
- Le Coroutine Builder
- Les fonctions “suspend”

7. Contrôler les coroutines

- Scope
- Context & Dispatchers
- Job & cancellation

8. Des coroutines qui communiquent

- Channels
- Producteur et consommateur

9. Traiter des flux avec les coroutines

- Flow
- CallbackFlow
- Opérations intermédiaires et terminales
- Composition
- shareIn
- Gestion d’erreur

10. Les génériques

- Déclarer un type générique
- Utiliser in, out, where
- Réifier un type générique

11. Délégation

- Héritage contre composition
- Déléguer une interface
- Déléguer une propriété

12. Réflexion

- KClass
- Les références de fonctions
- Les références de propriétés

13. Meta programmation

- Annotation processor
- Kotlin Symbol Processor
- Kotlin Compiler Plugins

14. Conclusion

03

Lieu, participants, prix

◆ 3.1 - Lieu de la formation

Cette formation sera assurée dans les locaux de 4SH, dans vos locaux ou à distance.

Pour les personnes en situation de handicap, nous mettrons tout en œuvre pour vous accueillir ou pour vous réorienter. Vous pouvez nous contacter au 09 63 28 62 73.

Si les conditions sanitaires ne permettent pas de réaliser cette formation sur site, celle-ci pourra être remplacée par une session à distance, aux mêmes dates.

◆ 3.2 - Participants

NOMBRE DE PARTICIPANTS PAR SESSION :

- Minimum : 1
- Maximum : 8

Inscription à réaliser 1 mois avant le démarrage de la formation.

◆ 3.3 - Durée et prix de la formation

DURÉE : 3 jours (soit 21h) par participant

HORAIRES : De 9h00 à 12h30 et de 14h00 à 17h30

PRIX : à partir de 2100€ HT par participant





Développez vos idées



www.4sh.fr