

# Kotlin – Mise en oeuvre

3 jours (21 heures)

Initiation

Présentiel ou classe à distance

à partir de

**1905 € HT** par personne (TVA en vigueur) **Repas inclus en présentiel**

Contenu mis à jour le 25/09/2026. Document téléchargé le 07/10/2026.

## Objectifs

A l'issue de cette formation, 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.

Les compétences visées par cette formation font l'objet d'une évaluation.

## À qui s'adresse cette formation ?

Développeurs et chefs de projets.

### Prérequis

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

## Programme

**1** Pendant votre formation

## Kotlin – Mise en oeuvre

Jour 1

### 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

*Exemple de travaux pratiques (à titre indicatif)*

- *Ecriture de divers algorithmes reprenant les principales syntaxes de Kotlin afin de concevoir une application simple*

### 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

*Exemples de travaux pratiques (à titre indicatif)*

- *A partir des projets préalablement générés, ajout et appels de plusieurs fonctions Kotlin*

### 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

*Exemple de travaux pratiques (à titre indicatif)*

- Création d'un projet classe / objet reprenant tous les principes de l'objet en Kotlin

Jour 2

## 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

*Exemple de travaux pratiques (à titre indicatif)*

- Création d'un programme de surcharge des opérateurs principaux pour les classes déjà créées

## 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

*Exemples de travaux pratiques (à titre indicatif)*

- *Ecriture de code permettant de mettre en lumière l'avantage de la délégation plutôt que l'héritage en Kotlin*
- *Ajout dans le code de Generics <>*
- *Ajout de la gestion des exceptions "try...catch" afin de rendre le programme plus solide*

Jour 3

## Interopérabilité

- Interopérabilité avec Java
- De Kotlin au Java
- Nulls de Java
- Le Kotlin dans Java
- Extensions de fonctions à partir du Java
- Interopérabilité avec les dernières versions de Java
- Java Réflexion avec Kotlin
- Kotlin Réflexion

*Exemple de travaux pratiques (à titre indicatif)*

- *Exemple d'intégration de code Java dans une application Kotlin et inversement*

## 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

*Exemple de travaux pratiques (à titre indicatif)*

- *Mise en oeuvre de l'asynchronisme et de la programmation réactive en Kotlin*

## Kotlin et l'ouverture sur Android

- Développer une application Android sur AndroidStudio en Kotlin

## Kotlin et l'IA

- Utiliser l'IA Générative pour concevoir des applications Kotlin
- Intégrer des API d'OpenAI dans Kotlin

*Exemples de travaux pratiques (à titre indicatif)*

- *Création rapide d'une application Android en Kotlin avec Android Studio*
- *Exemple d'utilisation d'outils d'IA Générative comme ChatGPT pour créer du code Kotlin adapté à l'application*

## Financement

Plusieurs dispositifs de financement sont accessibles via les OPCO (Opérateurs de Compétences), organismes agréés par le ministère du Travail dont le rôle est d'accompagner, collecter et gérer les contributions des entreprises au titre du financement de la formation professionnelle.

Pour plus d'information, une équipe de gestionnaires SKOLAE Formation spécialisée vous accompagne dans le choix de vos formations et la gestion administrative.

[Nous contacter pour en savoir plus →](#)

## Modalités pédagogiques

Formation délivrée en présentiel ou distanciel\* (blended-learning, e-learning, classe virtuelle, présentiel à distance).

Le formateur alterne entre méthode\*\* démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation).

Variables suivant les formations, les moyens pédagogiques mis en oeuvre sont :

- Ordinateurs Mac ou PC (sauf pour certains 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

En cas de formation intra sur site externe à M2I, le client s'assure et s'engage également à avoir toutes les ressources matérielles pédagogiques nécessaires (équipements informatiques...) au bon déroulement de l'action de formation visée conformément aux prérequis indiqués dans le programme de formation communiqué.

\* nous consulter pour la faisabilité en distanciel

\*\* ratio variable selon le cours suivi

### Accessibilité aux personnes en situation de handicap

## Détails de la session

Du 16 décembre 2026 au 18 décembre 2026 (3 jours / 21 heures)

Mulhouse

Du 16 décembre 2026 au 18 décembre 2026

Mulhouse

- 16 décembre 2026 ----- 9h00 – 16h00
- 17 décembre 2026 ----- 9h00 – 16h00
- 18 décembre 2026 ----- 9h00 – 16h00