

## Kotlin pour développeurs Java

### Développer des applications avec Kotlin

Avec le support officiel du langage Kotlin sur la plateforme Android, Google crée une véritable révolution chez les développeurs. Ce langage mature, créé en 2011 par la société JetBrains (l'auteur du très populaire IDE IntelliJ IDEA) amène tous les concepts et paradigmes modernes, objets et fonctionnels, avec un pragmatisme et une concision rafraichissante.

En l'officialisant sur la plateforme Android, Google propose un rival crédible au langage Swift, développé par Apple pour moderniser l'iOS. Techniquement proche de Java (Kotlin s'appuie sur la JVM), il promet du code plus concis (jusqu'à 30%), plus robuste et plus lisible que Java. Cette formation permettra aux développeurs, de préférence Java, de migrer rapidement vers ce langage d'avenir.

#### Détails

- **Code** : GO-KOTJ
- **Durée** : 4 jours ( 28 heures )

#### Public

- Consultants
- Consultants informatiques
- Développeurs

#### Pré-requis

- Connaître les bases de Java

#### Objectifs

- Découvrir le langage Kotlin
- Maîtriser les concepts et la syntaxe
- Migrer efficacement ses développeurs Java vers Kotlin

#### Programme

#### Historique

- Pourquoi Kotlin ?
- Les "défauts" de Java
- L'esprit Kotlin
- Installer l'environnement
- Créer son premier projet

#### Concepts et syntaxe

- Classe et Fonction
- Variable et propriétés
- Types de base
- Strings
- String template (interpolation)
- Package
- Modificateurs de visibilité
- Paramètre
- Valeur par défaut
- Arguments nommés
- Propriétés, getters et setters
- Constructeurs
- Le type Any
- Le type Unit
- if
- when
- Appels multiples (with)
- for
- while / do while
- ranges
- Gestion du null
- Test de type
- Conversion de type

- Comparaison

#### Concepts objets avancés

- Classe de donnée (data class)
- Héritage
- Redéfinition de méthode
- Redéfinition de propriétés
- Mot clé « super »
- Classes anonymes
- Singleton
- Opérateurs et surcharge
- Opérateurs disponibles
- Héritage et interface
- Exception
- Checked Exceptions ?
- Try, catch, finally...
- Le type Nothing
- Objet compagnon
- Interface et délégation
- Collections
- Type générique

#### Concepts fonctionnels

- Lambda
- Le mot clé « it »
- Valeur de retour
- Imbrications et labels
- Lambdas et Collection
- Opérations d'agrégation
- Opérations de mapping
- Opérations de filtrage

- Opérations d'éléments
- Opérations de génération
- Récursivité terminale (tailrec)
- Référence de fonction
- Closure
- Déclaration déstructurée
- Variable "underscore" ( \_ )

### Concepts avancés

- lateinit
- Delegated properties
- Lazy Properties
- Observable properties
- Alias
- Notation infixée
- Classes imbriquées
- Classes énumérées (enum)
- Classes scellée (sealed)

- Extension par décoration
- Lambdas avec receveur
- Librairie standard – let
- Librairie standard – run
- Librairie standard – also
- Librairie standard – apply
- Librairie standard – with
- Inline / Noinline
- Réflexion
- Annotation
- reified
- Builder
- Coroutine (expérimental)

### Intégration Java

- Avantages de Kotlin sur Java
- Appeler du code Java depuis Kotlin
- Appeler du code Kotlin depuis Java

### Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages