

# Formation Kotlin

Ihab ABADI / UTOPIOS

# SOMMAIRE

Partie 1 : Développement en Kotlin

Partie 2 : Utilisation du Kotlin pour le développement Android

# SOMMAIRE PARTIE 1

1. Déclaration de variable muable et mutable
2. Typage et Cast
3. Contrôle de flux
4. Package et auto-import
5. Kotlin et fonctions
6. Kotlin et objet
7. Opérateur de null safety
8. DSL
9. Opérateur de destruction
10. Programmation asynchrone

# SOMMAIRE PARTIE 2

1. Présentation de l'environnement Android
2. Structure de projet Android
3. Notion Activité
4. Manifest
5. Les principaux composants
6. Gestion des événements
7. Gestion des ressources
8. Gestion des droits
9. D'autres éléments
9. Programmation asynchrone android
10. Bibliothèques incontournables

# Variable muable et mutable

Kotlin permet la déclaration de variables:

- 1- Muable en lecture seul avec le mot clé val
- 2- Mutable en lecture et écriture avec le mot clé var

# Typage et cast

- Tout est objet en Kotlin.
- Kotlin possède également une représentation primitive de certain types (numbers, boolean, characters, strings)
- Kotlin possède any et null any
- Kotlin offre plusieurs opérateur pour vérifier le type d'une variable (is, as)

# Contrôle de flux (Structures)

- Structure if else
- Structure when
- Structure d'itération for
- Structure d'itération while
- Structure d'itération do while

# Contrôle de flux ( Retour et continue)

- Return: par défaut renvoie de la fonction englobante la plus proche ou de la fonction anonyme
- Break: termine la boucle englobante la plus proche
- Continue: passe à l'étape suivante de la boucle englobante la plus proche
- Toute expression dans Kotlin peut être marquée avec une étiquette.
- Les étiquettes ont la forme d'un identifiant suivi du signe @, par exemple : abc@, fooBar@.
- Pour étiqueter une expression, ajoutez simplement une étiquette devant elle.



# Package et auto-import

- Chaque élément en Kotlin peut être défini dans un package.
- Par défaut, dans Kotlin, dans chaque fichier un nombre de package et auto-importer.
- On peut importer un élément d'un package.
- On peut importer l'ensemble d'un package.

# Kotlin et fonctions

- Kotlin permet la programmation fonctionnelles.
- La création d'un Kotlin peut se faire à l'aide du mot clé fun.
- Le Kotlin permet de déclarer des fonctions en une seule expression.
- Le Kotlin permet de déclarer un nombre variable de paramètres.
- Le Kotlin permet de déclarer des fonctions Infix.

# Kotlin et fonctions

- Kotlin permet de créer des fonctions de Higher-order.
- Une fonction de Higher-order est une fonction qui accepte comme argument une fonction et qui renvoie une fonction.
- Kotlin permet de déclarer des fonctions anonymes et en une seule ligne avec les expressions lambda.

# Kotlin et Objets (Classes et constructeurs)

- Kotlin permet de faire de la POO.
- Kotlin permet de déclarer des classes
- Chaque classe en Kotlin peut avoir un constructeur primaire.
- Un constructeur primaire ne contient pas d'instruction.
- L'initialisation peut se faire à l'aide de la méthode init.
- Chaque classe en Kotlin peut avoir plusieurs constructeurs secondaires.
- Chaque constructeur secondaire doit invoquer le constructeur primaire si déclarer.

# Kotlin et Objets (Héritage)

- Kotlin permet de mettre en place de l'héritage.
- Kotlin permet surcharger des méthodes des classes parentes.

# Kotlin et Objets (Propriétés)

- Kotlin remplace le mécanisme de méthodes accesseurs par des propriétés.
- Chaque propriété peut définir une partie get et set

# Kotlin et Objets (Interface)

- Kotlin permet d'implémenter des interfaces.
- Les méthodes dans les interfaces peuvent avoir une implémentation par défaut.
- L'implémentation multiple est autorisée.
- Dans le cadre d'implémentation multiples, on peut avoir un conflit lors de la surcharge.
- Kotlin permet de résoudre ce conflit.
- Kotlin permet également d'implémenter des interfaces fonctionnelles.

# Kotlin et Objets (Visibilité des membres )

- Kotlin propose plusieurs niveau de visibilité.
- Private, protected, internal, et public



# Kotlin et Objets (Data class)

- Kotlin permet de créer des classes de types data qui ne contiennent pas de logique métier.
- Chaque data class possède une implémentation des méthodes equals, hashCode, toString

# Kotlin et Objets (générique et collections)

- Kotlin permet de mettre en place des génériques.
- Kotlin propose une multitude de collections.
- List, Set, Map
- Chaque collection possède une version muable et mutable.

# Opérateur de null safety

- Par défaut tout élément dans kotlin est non null.
- Dans le cadre ou on force à un élément à être nullable, on peut utiliser des opérateurs pour vérifier que l'élément est non null.
- Opérateur elvis
- Opérateur !!

# DSL

- Kotlin nous donne les outils pour créer du code en quelque chose qui semble plus naturel à utiliser, via un langage spécifique à un domaine - ou DSL
- Pour créer une DSL, on utilise le pattern du builder.
- Les expressions lambda
- Les fonctions de définition de scope

# Programmation asynchrone

- Kotlin possède plusieurs mécanisme pour implémenter une programmation asynchrone.
- Kotlin permet de créer des threads légers à l'aide de coroutine.
- La création de coroutine se fait à l'aide de la méthode launch
- Kotlin offre la possibilité de déclarer des fonctions qui s'exécutent dans une coroutine à l'aide du mot clé suspend
- Kotlin permet de lancer des coroutines également avec le mot clé async