

# Évaluation 4

## Les boucles

---



### Instructions

- Vous allez créer une application fictive pour gérer un festival de musique.
- Utilisez les concepts appris pour résoudre les exercices suivants.
- Stockez les résultats de chaque exercice dans une variable `resultats` et affichez-les à la fin avec `setContent { Text(resultats) }`.

### Programme du Festival

1. Créez une liste mutable `programme` avec les concerts suivants : "Rock Band", "Jazz Ensemble", "Electronic DJ".
2. Ajoutez "Classical Quartet" à la liste.
3. Retirez "Jazz Ensemble" de la liste.
4. Créez une chaîne `resultats` qui liste tous les concerts restants.
5. Affichez le contenu de `resultats`.

### Calcul des Billets Vendus

1. Créez une map mutable `billetsVendues` avec les concerts comme clés et les nombres de billets vendus comme valeurs. Exemple : "Rock Band" to 50.
2. Mettez à jour le nombre de billets vendus pour "Electronic DJ" avec une nouvelle valeur.
3. Utilisez une boucle `for` pour calculer le total des billets vendus.
4. Créez une chaîne `resultats` avec le nombre total de billets vendus.
5. Affichez le contenu de `resultats`.

### Vérification de Disponibilité des Artistes

1. Créez une map mutable `artistesConfirment` où les clés sont les noms des artistes et les valeurs sont des booleans indiquant leur présence (`true` pour confirmé, `false` pour non confirmé).
2. Ajoutez les artistes "Rock Band" (vérifié) et "Electronic DJ" (non vérifié) à la map.

3. Créez une chaîne `resultats` pour suivre l'état de confirmation.
4. Utilisez une boucle `while` pour vérifier la confirmation de chaque artiste et ajouter des messages à `resultats` jusqu'à ce que tous les artistes soient confirmés. Pour cela, si un artiste n'est pas confirmé, changez sa valeur pour qu'il le soit à la prochaine itération de la boucle.
5. Affichez le contenu de `resultats`.

## Gestion des Promotions pour les Billets

1. Création des Données :
  - Créez une map mutable `promotions` où les clés sont les noms des concerts et les valeurs sont des promotions potentielles (`String?`), ou `null` si aucune promotion n'est disponible.
  - Exemple initial : "Rock Band" to "10% de réduction", "Electronic DJ" to `null`.
2. Application des Promotions :
  - Créez une liste mutable `concerts` contenant les noms des concerts : "Rock Band", "Jazz Ensemble", "Electronic DJ".
  - Utilisez une boucle `for` pour parcourir chaque concert dans la liste `concerts`.
  - Pour chaque concert, récupérez la promotion de la map `promotions`.
  - Si une promotion est disponible (non-`null`), ajoutez-la à une chaîne `resultats` avec le message : "Promotion pour [concert] : [promotion]".
  - Si aucune promotion n'est disponible (`null`), ajoutez à `resultats` le message : "Aucune promotion pour [concert]".
3. Afficher les Résultats :
  - Utilisez `setContent { Text(resultats) }` pour afficher la chaîne `resultats` contenant les promotions pour chaque concert.

## Gestion des Artistes en Attente

1. Créez une liste mutable `artistesEnAttente` avec des noms d'artistes qui attendent encore une réponse.
2. Ajoutez quelques artistes à la liste.
3. Utilisez une boucle `do while` pour traiter chaque artiste en ajoutant un message de confirmation à `resultats` et en les retirant de la liste.
4. Affichez le contenu de `resultats`.

## Répartition des Tâches

1. Créez une liste mutable `taches` avec les tâches suivantes : "Monter scène", "Vérifier sonorisation", "Accueillir les artistes".
2. Créez une map mutable `repartitionTaches` où les clés sont les noms des membres de l'équipe et les valeurs sont des listes de tâches assignées.
3. Assignez les tâches aux membres de l'équipe.
4. Utilisez une boucle `for` pour afficher les tâches assignées à chaque membre.
5. Créez une chaîne `resultats` avec la répartition des tâches et affichez-la.