Les Maps en Dart (POO)

Objectifs

- Comprendre le concept des Maps en Dart
- Utiliser les Maps pour gérer des paires clé-valeur
- Manipuler des Maps dans un contexte orienté objet

1. ## Présentation
    - En Dart, une Map est un objet de collection dynamique qui associe des clés (keys) à des valeurs (values). Chaque clé doit être unique, mais les valeurs peuvent être dupliquées.
    - Bien qu'on les utilise souvent de manière littérale, les Maps en Dart s'inscrivent pleinement dans le paradigme de la Programmation Orientée Objet (POO). Ce sont des instances de la classe générique Map<K, V>.
2. ## Déclaration et Initialisation
    - Puisque `Map` est une classe, vous pouvez créer une instance de plusieurs manières :
    - Déclaration littérale :

```
Map<String, int> notes = {
  'Ali': 14,
  'Sami': 16,
  'Maya': 18
};
```

3. Avec constructeur :
4. En tant que classe, `Map` propose plusieurs constructeurs d'usine (factory constructors) pour des besoins spécifiques :

```
// Constructeur par défaut (crée une Map vide)
var scores = Map<String, int>();

// Création d'une Map immuable (non modifiable)
var configuration = Map<String, String>.unmodifiable({
  'api_url': 'https://api.exemple.com',
  'timeout': '30s',
});
```

apcpedagogie.com

```
// Création à partir d'itérables (Listes)
Listélt:String> cles = ['Rouge', 'Vert', 'Bleu'];
List<String> valeurs = ['#FF0000', '#00FF00', '#0000FF'];
var codeCouleurs = Mapt<String, String>.fromIterables(cles, valeurs);
```

5. **Exemple:**

```
class Utilisateur {
  final String nom;
  final String role;
  Utilisateur(this.nom, this.role);

  @override
  String toString() => '$nom ($role)';
}

void main() {
  List<int> ids = [101, 102, 103];

  List<Utilisateur> utilisateurs = [
    Utilisateur('Alice', 'Admin'),
    Utilisateur('Bob', 'Manager'),
    Utilisateur('Charlie', 'Développeur'),
  ];

  // Fusion des deux listes en une Map indexée par ID
  Mapt<int, Utilisateur> annuaire = Map.fromIterables(ids, utilisateurs);

  print(annuaire[102]); // Affichera : Bob (Manager)

for (var entree in annuaire.entries) {
  print('ID: ${entree.key} -> Utilisateur: ${entree.value}');
}
}
```

6. Opérations sur les Maps

- Accès à une valeur : `map['clé']`
- Ajout/modification : `map['clé'] = valeur;`
- Suppression : `map.remove('clé');`
- Parcours :

apcpedagogie.com

```
void main() {
  // 1. Création de la Map (Clé: Nom, Valeur: Note)
  Map<String, double> notes = {
    'Alice': 18.5,
    'Bob': 14.0,
    'Charlie': 11.5,
  };

  // 2. Exemple de parcours avec .forEach()
  notes.forEach((nom, note) {
    print('$nom a obtenu $note');
  });
}
```

1. **Accès à une valeur**

```
print(personne['nom']); // Résultat : Ahmed
```

   - Utilisez la clé entre crochets pour obtenir la valeur correspondante.

2. **Ajout ou modification d'une valeur**

```
personne['profession'] = 'Développeur';
personne['ville'] = 'Sfax';
```

   - Si la clé n'existe pas, elle est ajoutée.
   - Si elle existe déjà, la valeur est modifiée.

3. **Suppression d'une entrée**

```
personne.remove('âge');
```

   - Supprime l'entrée correspondant à la clé `'âge'`.

4. **Parcours d'une Map**

   - **Parcours avec `forEach`:**

apcpedagogie.com

```
personne.forEach((cle, valeur) {
  print("$cle : $valeur");
});
```

5. **Parcours avec une boucle `for-in`:**

```
for (var cle in personne.keys) {
  print("$cle → ${personne[cle]}");
}
```

6. **Vérification d'une clé ou d'une valeur**

```
print(personne.containsKey('nom')); // true
print(personne.containsValue('Sfax')); // true
```

- `containsKey`: vérifie si une clé existe
- `containsValue`: vérifie si une valeur est présente

7. **Obtenir les clés et les valeurs**

```
print(personne.keys);   // (nom, ville, profession)
print(personne.values); // (Ahmed, Sfax, Développeur)
```

8. **Vider complètement une Map**

```
personne.clear();
```

- Supprime toutes les paires clé/valeur de la Map.

9. **Activité 01**

- Crée un fichier Dart nommé map_operations.dart. Tu vas créer et manipuler une Map représentant un étudiant.
- Déclare une Map nommée etudiant contenant les informations suivantes :
  nom : 'Ali' age : 21 specialite : 'Informatique'
- Affiche le nom de l'étudiant.
- Ajoute une nouvelle clé note avec la valeur 16.5.

apcpedagogie.com

- Modifie la spécialité en 'Cybersécurité'.
- Supprime l’âge de la Map.
- Affiche tous les éléments de la Map en utilisant une boucle forEach.
- Teste si la clé note existe. Si oui, affiche : "Note trouvée".
- Vide complètement la Map.

  Solution

```
void main() {
  // 1. Déclaration de la Map
  Map<String, dynamic> etudiant = {
    'nom': 'Ali',
    'age': 21,
    'specialite': 'Informatique'
  };

  // 2. Afficher le nom de l'étudiant
  print('Nom : ${etudiant['nom']}');

  // 3. Ajouter une nouvelle clé 'note' avec la valeur 16.5
  etudiant['note'] = 16.5;

  // 4. Modifier la spécialité en 'Cybersécurité'
  etudiant['specialite'] = 'Cybersécurité';

  // 5. Supprimer l’âge de la Map
  etudiant.remove('age');

  // 6. Afficher tous les éléments de la Map avec une boucle forEach
  print('Contenu de la Map :');
  etudiant.forEach((cle, valeur) {
    print('$cle : $valeur');
  });

  // 7. Tester si la clé 'note' existe
  if (etudiant.containsKey('note')) {
    print('Note trouvée');
  }

  // 8. Vider complètement la Map
  etudiant.clear();

  // Afficher la Map vide
```

apcpedagogie.com

```
  print(etudiant); // {}
}
```

10. **Activité02**

- Crée une Map<String, Map<String, dynamic>> nommée classe contenant 3 étudiants avec les données suivantes :

  // Exemple de structure d'un étudiant dans la Map : 'nom': 'Ali', 'age': 21, 'specialite': 'Informatique', 'notes': [15.0, 16.5, 17.0]
- Affiche les informations complètes de chaque étudiant à l'aide d'une boucle forEach.
- Calcule et affiche la moyenne des notes de chaque étudiant.
- Ajoute un nouvel étudiant Salma à la Map.
- Modifie la spécialité de Ali en Cybersécurité.
- Supprime l'étudiant Omar de la classe.
- Affiche uniquement les étudiants qui ont une moyenne supérieure à 16.
- Vide complètement la Map.

  Solution

```
void main() {
  // Étape 1 : Créer la Map classe avec 3 étudiants
  Map<String, Map&gtString, dynamic>> classe = {
    'Ali': {
      'age': 21,
      'specialite': 'Informatique',
      'notes': [15.0, 16.5, 17.0]
    },
    'Omar': {
      'age': 22,
      'specialite': 'Mathématiques',
      'notes': [13.5, 14.0, 12.5]
    },
    'Sami': {
      'age': 20,
      'specialite': 'Physique',
      'notes': [17.5, 18.0, 19.0]
    }
  };

  // Étape 2 : Afficher toutes les informations
```

apcpedagogie.com

```
  print('Informations des étudiants :');
  classe.forEach((nom, infos) {
    print('Nom : $nom');
    infos.forEach((cle, valeur) {
      print('  $cle : $valeur');
    });
  });

  // Étape 3 : Calculer et afficher les moyennes
  print('\nMoyennes :');
  classe.forEach((nom, infos) {
    List<double> notes = List<double>.from(infos['notes']);
    double moyenne = notes.reduce((a, b) => a + b) / notes.length;
    print('$nom : $moyenne');
  });

  // Étape 4 : Ajouter un nouvel étudiant
  classe['Salma'] = {
    'age': 23,
    'specialite': 'Informatique',
    'notes': [16.0, 17.0, 15.5]
  };

  // Étape 5 : Modifier la spécialité de Ali
  classe['Ali']?['specialite'] = 'Cybersécurité';

  // Étape 6 : Supprimer l'étudiant Omar
  classe.remove('Omar');

  // Étape 7 : Afficher les étudiants avec moyenne > 16
  print('\nÉtudiants avec moyenne > 16 :');
  classe.forEach((nom, infos) {
    List notes = List<double>.from(infos['notes']);
    double moyenne = notes.reduce((a, b) => a + b) / notes.length;
    if (moyenne > 16) {
      print('$nom avec une moyenne de $moyenne');
    }
  });

  // Étape 8 : Vider la classe
  classe.clear();
  print('\nClasse après suppression : $classe');
}
```

apcpedagogie.com

7. Map dans une classe (POO)

- Exemple d'utilisation orientée objet :

```
class Etudiant {
  String nom;
  Map<String, double> notes = {};

  Etudiant(this.nom);

  void ajouterNote(String matiere, double note) {
    notes[matiere] = note;
  }

  double moyenne() {
    double total = notes.values.reduce((a, b) => a + b);
    return total / notes.length;
  }

  void afficherNotes() {
    print('Notes de $nom :');
    notes.forEach((matiere, note) {
      print('- $matiere : $note');
    });
  }
}

void main() {
  var e = Etudiant('Amine');
  e.ajouterNote('Math', 15);
  e.ajouterNote('Physique', 14);
  e.afficherNotes();
  print('Moyenne : ${e.moyenne()}');
}
```

8. Méthodes utiles avec les Maps

- `.containsKey()`: vérifie la présence d'une clé
- `.keys` et `.values`: accède aux clés ou valeurs
- `.clear()`: vide la Map
- `.update()` et `.putIfAbsent()` pour modifier ou insérer

apcpedagogie.com

9. ## Map imbriquée

   - Exemple :

```
Map<String, Map<String, double>> bulletin = {
  'Ali': {'Math': 14, 'SVT': 16},
  'Maya': {'Math': 18, 'SVT': 15}
};
```

10. Accès :

```
print(bulletin['Ali']?['Math']); // Affiche 14
```

Cours Dart

Source : https://apcpedagogie.com/les-maps-en-dart-poo · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com