

Leçon 13

Les Tableaux (Array)



Objectif

Dans cette leçon, vous allez découvrir ce qu'est un tableau ([array](#)), comment en créer un, comment accéder à ses éléments et comprendre ses limitations. Les tableaux sont utiles pour stocker plusieurs valeurs du même type dans une seule variable.

Qu'est-ce qu'un Tableau ?

Un tableau est une collection ordonnée d'éléments du même type. Imaginez une boîte à compartiments où chaque compartiment contient un objet. Par exemple, une boîte à crayons où chaque compartiment contient un crayon différent.

Comment Créer un Tableau

Pour créer un tableau en Kotlin, vous utilisez l'instruction [arrayOf\(\)](#) et vous placez les éléments que vous voulez stocker entre parenthèses, séparés par des virgules.

Syntaxe :

```
val nomDuTableau = arrayOf(élément1, élément2, élément3, ...)
```

- [val](#) : Le mot-clé qui signifie que vous créez une variable.
- [nomDuTableau](#) : Le nom que vous donnez à votre tableau.
- [arrayOf](#) : L'instruction utilisée pour créer un tableau.
- [élément1](#), [élément2](#), [élément3](#) : Les éléments que vous voulez stocker dans le tableau, séparés par des virgules.

Exemple :

Supposons que vous ayez trois fruits : une pomme, une banane et une cerise. Au lieu de créer une variable pour chaque fruit, vous pouvez les regrouper dans un tableau.

```
val fruits = arrayOf("Pomme", "Banane", "Cerise")
```

Ici, nous avons créé un tableau appelé `fruits` qui contient trois éléments : "Pomme", "Banane", et "Cerise". Chaque élément du tableau est de type `String`.

Accéder aux Éléments d'un Tableau

Pour accéder à un élément spécifique du tableau, vous utilisez son index (le numéro du compartiment).

Attention, l'indexation commence à 0, donc le premier élément se trouve à l'index 0, le deuxième à l'index 1, et ainsi de suite.

Exemple :

Pour accéder à la "Banane" dans le tableau :

```
val fruits = arrayOf("Pomme", "Banane", "Cerise")  
val fruitFavori = fruits[1] // "Banane"
```

Taille d'un Tableau : Une Limitation Importante

Un point essentiel à comprendre avec les tableaux est que leur taille est fixée lors de leur création. Imaginez que votre boîte à crayons a trois compartiments. Vous ne pouvez pas ajouter un quatrième compartiment ni en supprimer un. Vous pouvez seulement changer ce qu'il y a dans chaque compartiment.

Exemple :

Si vous créez un tableau avec trois fruits, vous ne pourrez pas en ajouter un quatrième plus tard, ni en retirer un. Vous pouvez seulement modifier les fruits existants.

```
val fruits = arrayOf("Pomme", "Banane", "Cerise")  
// Vous ne pouvez pas faire fruits[3] = "Fraise" car le tableau a seulement 3 compartiments
```

Pourquoi c'est important ?

Cette limitation des tableaux est importante à connaître. Si vous avez besoin d'une collection où vous pouvez ajouter ou supprimer des éléments, il faudra utiliser un autre type de structure de données, comme une liste mutable (que nous verrons à la leçon suivante).

Longueur d'un tableau

Pour savoir combien d'éléments il y a dans un tableau, vous pouvez utiliser la propriété `size` du tableau.

Exemple :

Pour connaître le nombre de fruits dans le tableau :

```
val fruits = arrayOf("Pomme", "Banane", "Cerise")
val nombreDeFruits = fruits.size // 3
```

Modifier les Éléments d'un Tableau

Comme vu précédemment, vous ne pouvez pas ajouter ou supprimer des éléments d'un tableau une fois qu'il a été créé, mais vous pouvez modifier les éléments existants en utilisant leurs index.

Exemple :

Si vous voulez remplacer "Cerise" par "Orange" dans le tableau :

```
val fruits = arrayOf("Pomme", "Banane", "Cerise")
fruits[2] = "Orange"
```

Maintenant, le tableau `fruits` contient ["Pomme", "Banane", "Orange"].

Manipulations Utiles avec un Tableau

1. Accès au Premier Élément

Le premier élément d'un tableau est toujours accessible via l'index 0. C'est comme ouvrir la première boîte de votre collection.

Exemple :

Pour accéder au premier fruit dans le tableau :

```
val premierFruit = fruits[0] // "Pomme"
```

Il existe aussi une manière plus simple et intuitive avec la propriété `first()` :

```
val premierFruit = fruits.first() // "Pomme"
```

2. Accès au Dernier Élément

Pour accéder au dernier élément d'un tableau, vous pouvez utiliser l'index `size - 1`, car `size` donne le nombre total d'éléments, mais l'index commence à 0.

Exemple :

Pour accéder au dernier fruit du tableau :

```
val dernierFruit = fruits[fruits.size - 1] // "Cerise"
```

Il existe aussi une manière plus simple et intuitive avec la propriété `last()` :

```
val dernierFruit = fruits.last() // "Cerise"
```

3. Recherche d'un Élément

Vous pouvez vérifier si un tableau contient un certain élément en utilisant l'opérateur `in` ou la méthode `contains()`.

Exemple :

Pour vérifier si "Banane" est dans le tableau :

```
val aUneBanane = "Banane" in fruits // true
```

ou bien :

```
val aUneBanane = fruits.contains("Banane") // true
```

Exercices

Exercice 1

1. Créez un tableau nommé `fruits` contenant les éléments suivants : "Pomme", "Banane", "Cerise", "Raisin".
2. Affichez le contenu du tableau `fruits` sous la forme suivante :
"Le tableau contient : Pomme, Banane, Cerise, Raisin."

Exercice 2

1. Créez un tableau nommé `animaux` contenant les éléments suivants : "Chien", "Chat", "Lapin", "Oiseau".
2. Affichez le premier élément du tableau `animaux` sous la forme :
"Le premier animal est : Chien."

Exercice 3

1. Créez un tableau nommé pays contenant les éléments suivants : "France", "Espagne", "Italie", "Allemagne".
2. Affichez le dernier élément du tableau pays sous la forme :
"Le dernier pays est : Allemagne."

Exercice 4

1. Créez un tableau nommé couleurs contenant les éléments suivants : "Rouge", "Vert", "Bleu", "Jaune".
2. Remplacez "Vert" par "Violet" dans le tableau couleurs.
3. Affichez le contenu du tableau après la modification :
"Le tableau modifié contient : Rouge, Violet, Bleu, Jaune."

Exercice 5

1. Créez un tableau nommé jours contenant les éléments suivants : "Lundi", "Mardi", "Mercredi", "Jeudi".
2. Créez une variable recherche de type String avec la valeur "Mercredi". Utilisez cette variable pour vérifier si "Mercredi" est présent dans le tableau jours.
3. Affichez un texte indiquant si "Mercredi" est trouvé :
"Mercredi est présent dans le tableau."

Exercice 6

1. Créez un tableau nommé saisons contenant les éléments suivants : "Printemps", "Été", "Automne", "Hiver".
2. Affichez un texte indiquant combien de saisons sont stockées dans le tableau saisons :
"Le tableau contient 4 saisons."