# Tableaux

Un tableau est une **structure de données** qui permet de stocker un ensemble d'éléments sous une même variable. Les tableaux sont utilisés dans presque tous les langages de programmation pour organiser et manipuler des collections de données de manière efficace.

## Principes

Les tableaux reposent sur l'idée que les éléments peuvent être stockés de manière proche en mémoire. Chaque élément du tableau occupe une **position spécifique**, ce qui permet d'y accéder rapidement en utilisant un **indice**. Les tableaux sont souvent utilisés pour stocker des séries de **valeurs** de même type

## Usages

Les tableaux sont utilisés pour :

- **La gestion de collections d'éléments** : Un tableau permet de regrouper des données similaires, ce qui facilite les opérations de recherche, de tri, d'addition, de suppression, etc.

- **L'optimisation de la performance** : Les tableaux offrent un accès rapide aux éléments grâce à l'utilisation d'indices.

- **La manipulation de structures complexes** : Les tableaux multidimensionnels sont largement utilisés pour représenter des matrices, des tableaux de pixels, des réseaux ou des bases de données.

### Tableaux indexés

Un tableau indexé est un type de tableau où chaque élément est accessible via un indice, généralement un entier. Les indices dans un tableau sont souvent basés sur des entiers séquentiels (0, 1, 2, ...).

#### Déclaration

Dans de nombreux langages de programmation, la déclaration d'un tableau indexé consiste à spécifier le type des éléments ainsi que la taille ou l'initialisation des valeurs.

Exemple en Python:

```
tableau = [1, 2, 3, 4, 5]
```

#### Notation et usage

Les tableaux indexés sont utilisés principalement pour accéder à des éléments par leur indice. L'indexation commence généralement à 0. L'accès à un élément du tableau se fait par la notation suivante : tableau[indice] Par exemple, pour accéder à l'élément d'indice 2 dans un tableau Python :

```
print(tableau[2])
```

#### Parcours

Les tableaux indexés peuvent être parcourus à l'aide de boucles (comme for ou while). En voici un exemple en Python :

```
for i in range(len(tableau)):
    print(tableau[i])
```

### Tableaux associatifs

Un tableau associatif est une **structure de données** où chaque élément est accessible via une clé unique. Contrairement aux tableaux indexés qui utilisent des indices numériques, les tableaux associatifs utilisent des clés qui peuvent être de n'importe quel type (chaîne de caractères, nombre, etc.).

## Tableaux à plusieurs dimensions

Un tableau à plusieurs dimensions est un tableau qui contient d'autres tableaux. Cela permet de représenter des structures de données plus complexes.

#### Déclaration

La déclaration d'un tableau à plusieurs dimensions dépend du langage utilisé. Par exemple :

```
tableau = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

#### Notation et usage

Les tableaux à plusieurs dimensions sont accessibles en utilisant plusieurs indices, un pour chaque dimension. Par exemple, pour accéder à l'élément en ligne 1, colonne 2 dans un tableau 2D en Python :

```
print(tableau[1][2])
```

### Parcours

Les tableaux à plusieurs dimensions peuvent être parcourus de manière imbriquée avec des boucles. Voici un exemple en Python pour parcourir un tableau :

```
for ligne in tableau:
    for element in ligne:
        print(element, end=" ")
    print(
```

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
**https://wiki.siochaptalqper.fr/doku.php?id=bloc1:prog:tableaux&rev=1731012715**

Last update: **2024/11/07 21:51**