

## Traitement de données en tables

Cette fiche de révision appartient au chapitre « Traitement de données en tables ». Les notions suivantes sont abordées : comment importer un table, propriétés et manipulations d'un tableau.

### Comment importer un table :

Pour stocker les données en table on peut utiliser un logiciel « **tableur** » ou directement dans un fichier. Le format **csv** (données séparés par des virgules ou un autre délimiteur) et pratique pour stocker ce type de données, on appelle cela des **fichiers plats**.

Exemple d'une ligne dans un fichier *csv*: *FRANCE;NSI;1ère;2020;tableur;19;ok*

Il est possible d'utiliser un fichier *csv* comme donnée d'entrée d'un programme, comme donnée de sortie ou les deux à la fois.

Pour importer un fichier *csv* en Python

- On importe simplement un fichier en mode lecture 'r' ou lecture et écriture 'rw'  
*mon\_fichier = open("test.csv", "r")*
- On l'importe en utilisant les fonctions natives de Python  
*import csv*  
*with open('test.csv', newline='') as csvfile:*

Dans tous les cas on pourra avoir accès à un ou plusieurs champs, faire des ajouts, des suppressions, des tris car on manipulera des données de type liste.

Dans ce dernier cas il est possible d'utiliser une bibliothèque Python: **Pandas**, qui est plus adaptée pour le traitement de données en table:

```
import pandas  
pays = pandas.read_csv("test.csv", delimiter=";")
```

Dès lors nous pouvons utiliser des fonctions natives de cette bibliothèque spécifique:

- *pays.head()*: pour afficher les premiers **tuples** (éléments) de la table
- *pays.sample(5)*: pour afficher une sélection de 5 tuples au hasard de la table
- *pays.columns*: pour retourner la liste des champs
- *pays.dtype*: pour retourner la liste des champs accompagnés de leurs types d'objets
- *pays.loc[5]*: affichera le 5ème enregistrement du tableau
- *pays.loc[5]['nom']*: affichera 'Belgique'
- *pays[devise == 'EUR']*: retournera tous les tuples qui ont une devise en Euro

Dans le monde professionnel, pour de gros volumes on utilisera des **SGBD** (Système de Gestion de Bases de données) et on créera des bases de données composées de tables que l'on pourra interroger ou modifier avec le langage **SQL**: *SELECT \* FROM PANIER WHERE nom like 'AIR FORCE'*

### Propriétés et manipulations d'un tableau:

Dans cette fiche on utilisera les tableaux en langage Python (langage référence du programme)  
Un tableau est une suite ordonnée d'éléments. Ces éléments sont séparés par des virgules et

entourés par des crochets.

En **Python**, un tableau est de type **list**. Il peut être composé :

- d'entiers: `t1 = [1, 2, 9]`
- de chaînes de caractères: `t2 = ['toi', 'moi', 'nous']`
- de tuples `t3 = [(1, 2, 'moi'),(3, 4, 'toi'),(5, 6, 'nous')]`
- de tableaux `t4 = [[0, 2, 4, 6, 8],[1, 3, 5, 7, 9]]`

Le premier élément d'un tableau commence à l'indice 0.

(C'est le cas en Python cela peut-être 1 dans d'autres langages)

- `t2[0]` retournera 'toi'
- `t4[0][1]` retournera 3
- `t1[1:]` retournera le tableau de l'indice numéro 1 jusqu'à la fin soit: [2, 9]
- `t1[:1]` retournera le tableau de l'indice 0 jusqu'à l'indice 1 soit: [1,2]

Un tableau est **mutable** (on dit aussi muable), c'est à dire que l'on peut modifier l'un de ses éléments: si on affecte 9 au premier élément du tableau `t1`: `t1[0] = 9` alors `t1` retournera [9, 2, 9]

Attention : Deux tableaux qui sont liés par une relation d'égalité pointent vers la même adresse mémoire. La modification de l'un, entraîne la modification de l'autre.

Pour faire une vraie copie de tableau on peut utiliser la fonction `deepcopy()` du module `copy`.

Les opérateurs `+` et `*` s'utilisent comme avec les chaînes de caractères et les tuples. Ce sont des opérateurs de **concaténation** (désigne l'action de mettre bout à bout au moins deux chaînes de caractères) .

Il existe de nombreuses fonctions définies pour les tableaux, par exemple:

- **len()** pour connaître la longueur d'un tableau : `len(t1)` retournera 3
- **append()** pour ajouter un élément à le fin du tableau: `t1.append(4)` retournera [1, 2, 9, 4]
- **sort()** pour trier un tableau: `t1.sort()` retournera [1, 2, 4, 9]
- **insert()** pour insérer à l'indice d'un tableau: `t1.insert(0,10)` retournera [10, 1, 2, 4, 9]
- **reverse()** pour retourner le tableau: `t1.reverse()` retournera [9, 4, 2, 1, 10]
- **index()** pour retourner l'indice d'un élément du tableau: `t1.index(3)` retournera 1
- **remove()** pour supprimer un élément du tableau: `t1.remove(2)` retournera [9, 4, 1, 10]

## CE QU'IL FAUT RETENIR

En informatique on traite une quantité énorme de données, il faut les stocker en tables dans des fichiers ou dans des bases de données. Il est ainsi facile d'avoir accès a une donnée particulière (identifier par un critère) ou toutes les données. Il est aussi facile de manipuler ces données.