



LANGAGE PYTHON : MANIPULATION DE TABLEAUX

1 – TABLEAUX ET MATRICES

Une liste peut être composée de différents types et notamment d'autres listes. Une liste composée de listes représente un tableau (ou une matrice).

In []:

```
tableau = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
for ligne in tableau :
    print(ligne)
```

Par exemple, une image peut être représentée au moyen d'une table constituée de n lignes et de p colonnes où chaque élément représente un pixel.

Pour construire une table, il est possible d'utiliser, à partir d'une table vide, une boucle **for** qui va créer les lignes une à une. Par exemple :

In []:

```
table = [] # Création d'une table vide
for n in range(4) :
    ligne = [4*n+i for i in range(1, 5)]
    print(ligne)
    table.append(ligne)
```

Pour accéder aux différents éléments :

In []:

```
table[0] # accès à la 1ère ligne
```

In []:

```
table[3] # accès à la dernière ligne
```

In []:

```
table[3][3] # accès au dernier élément de la dernière ligne
```

In []:

```
table[3][4] # accès à un élément n'existant pas
```

2 – MANIPULATION DE TABLEAUX

2.1 – Présentation du format CSV

Le format **CSV** (pour *comma separated values*, soit en français *valeurs séparées par des virgules*) est un format informatique permettant de **stocker des tableaux de données dans un fichier texte**. Les règles du format CSV sont :

- Chaque ligne du fichier correspond à une ligne du tableau. Les valeurs de chaque colonne du tableau sont séparées par un caractère de séparation, en général une virgule ou un point-virgule. Chaque ligne est terminée par un caractère de fin de ligne (line break).
- Toutes les lignes contiennent obligatoirement le même nombre de valeurs (donc le même nombre de caractères de séparation). Les valeurs vides doivent être exprimées par deux caractères de séparation contigus.
- La taille du tableau est le nombre de lignes multiplié par le nombre de valeurs dans une ligne.
- La première ligne du fichier peut être utilisée pour exprimer le nom des colonnes (champs).

Par exemple, le tableau ci-dessous, contient la liste (extrait) des prénoms attribués aux enfants nés en France hors Mayotte entre 1900 et 2018 et les effectifs par sexe et département associés à chaque prénom.

sexe	preusuel	annais	dpt	nombre
1	AADIL	1983	84	3
1	AADIL	1992	92	3
1	AAHIL	2016	95	3
1	AARON	1962	75	3
1	AARON	1976	75	3
1	AARON	1982	75	3
1	AARON	1984	75	3
1	AARON	1985	75	5
1	AARON	1989	75	3
1	AARON	1990	69	3
1	AARON	1990	75	4
1	AARON	1990	93	4

Le fichier CSV contenant les données du tableau précédent, contient les lignes suivantes :

```
sexe;preusuel;annais;dpt;nombre
1;AADIL;1983;84;3
1;AADIL;1992;92;3
1;AAHIL;2016;95;3
1;AARON;1962;75;3
1;AARON;1976;75;3
1;AARON;1982;75;3
1;AARON;1984;75;3
1;AARON;1985;75;5
1;AARON;1989;75;3
1;AARON;1990;69;3
1;AARON;1990;75;4
1;AARON;1990;93;4
```

Le format CSV est particulièrement utilisé dans l'**Open Data** (données ouvertes).

L'open data, c'est quoi ? (<https://vimeo.com/266308637>) Il s'agit de données auxquelles tout le monde peut accéder et que tout le monde peut utiliser et partager. Les gouvernements, les entreprises et les individus peuvent utiliser l'open data afin de créer des avantages sociaux, économiques et environnementaux.

Ces données sont d'origine publique (documents administratifs comme les commandes publiques, données législatives, données cartographiques...) ou privée (papiers de recherche, administration d'entreprises, particuliers et crowdsourcing ...).

Un certain nombre de sites internet permettent d'accéder à ces données, parmi lesquels on peut citer : [data.gouv](https://data.gouv.fr) (<https://data.worldbank.org/>), [INSEE](https://www.insee.fr/fr/statistiques?categorie=1) (<https://www.insee.fr/fr/statistiques?categorie=1>), [World Bank Open data](https://data.worldbank.org/) (<https://data.worldbank.org/>)...

Le fichier **nat2018.csv** qui se trouve dans le répertoire ressource contient les prénoms attribués aux enfants nés en France hors Mayotte entre 1900 et 2018 et les effectifs par sexe associés à chaque prénom.

sexe	preusuel	annais	nombre
1	ABDON	1900	4
1	ABEL	1900	428
1	ABRAHAM	1900	19
1	ACHILLE	1900	205
1	ACHILLES	1900	6
1	ADALBERT	1900	7
1	ADAM	1900	20
1	ADELIN	1900	7
1	ADHEMAR	1900	5
1	ADOLPHE	1900	495
1	ADONIS	1900	22

```
sexe;preusuel;annais;nombre
1;ABDON;1900;4
1;ABEL;1900;428
1;ABRAHAM;1900;19
1;ACHILLE;1900;205
1;ACHILLES;1900;6
1;ADALBERT;1900;7
1;ADAM;1900;20
1;ADELIN;1900;7
1;ADHEMAR;1900;5
1;ADOLPHE;1900;495
1;ADONIS;1900;22
```

Le tableau de données contenu dans ce fichier est constitué de 4 colonnes :

- **sexe** : correspond au sexe de l'enfant - 1 pour les garçons et 2 pour les filles.
- **preusuel** (prénom usuel) : prénom donné à l'enfant.
- **annais** (année de naissance).
- **nombre** : nombre d'enfants auquel ce prénom a été donné.

Question 1 : Lancez un éditeur de texte. Ouvrir le fichier **nat2018.csv**. Recherchez dans les données, votre prénom et votre année de naissance. Précisez si cette recherche est facile.

Question 2 : Ouvrez à nouveau le fichier **nat2018.csv** avec un tableur. Donnez le nombre de lignes qui sont contenues dans le tableau.

Question 3 : Placez le curseur sur une case du tableau. Activez la fonction **Filtrer** (Menu **Données**).

Question 4 : Appliquez un filtre sur les prénoms afin de ne garder que votre prénom. Donnez le nombre d'enfants nés la même année que vous et qui portent votre prénom.

Question 5 : Déterminez le nombre d'enfants à qui on a donné le même prénom que vous depuis 1900.

2.2 – Importation d'un tableau

Pour charger un fichier CSV en python et lire les données, il faut utiliser la fonction **open** et la méthode **reader** disponibles dans le module **csv**.

Le code ci-dessous permet d'ouvrir le fichier **nat2018.csv** et de lire les données dans le tableau **tab_donnees**.

In []:

```
import csv # importation de la librairie csv

tab_donnees = [] # Création du tableau donnees vide

with open('nat2018.csv', newline='') as csvfile : # Ouverture du fichier nat2018.csv
    donnees = csv.reader(csvfile, delimiter=';') # Lecture des données
    for ligne in donnees : # Lecture des lignes de données et affectation au tableau lignes
        tab_donnees.append(ligne)

tab_donnees
```

Avec le code précédent, les données sont stockées dans un tableau de tableaux. Il n'y a pas de lien direct entre les champs (noms des colonnes, c'est à dire la première ligne du tableau **tab_donnees**) et le reste des données.

Pour remédier à cela, la méthode **DictReader** retourne un chaque dictionnaire pour chaque ligne du fichier CSV.

In []:

```
import csv # importation de la librairie csv
tab_donnees = [] # Création du tableau donnees vide
with open('nat2018.csv', newline='') as csvfile : # Ouvverture du fichier nat2018.csv
    donnees = csv.DictReader(csvfile, delimiter= ';') # Lecture des données
    for ligne in donnees : # Lecture des lignes de données et affectation au tableau lignes
        tab_donnees.append(dict(ligne))
tab_donnees
```

Cette fois-ci on obtient un tableau constitué de dictionnaires correspondant à chaque des lignes du fichier CSV et dont les clés sont les champs du tableau de données.

Les données enregistrées dans un fichier CSV sont des données texte, donc des données de type **str**. Il peut être souvent nécessaire d'adapter le type de certaines données (entiers, flottants) pour pouvoir les traiter.

Question 6 : Modifier le code précédent afin que les valeurs de la colonne **nombre** soient du type entier **int**.

In []:

2.2 – Recherche dans un tableau

Il est possible de récupérer des données du tableau à partir des différentes clés. Le code ci-dessous permet d'afficher, pour chacune des années depuis 1900, le nombre de filles ayant reçu le prénom "ANNE".

In []:

```
for prenom in tab_donnees :
    if prenom['preusuel'] == 'ANNE' and prenom['sexe'] == '2' :
        print(prenom['annais'], prenom['nombre'])
```

Le code ci-dessous permet d'afficher le nombre total de fois qu'une fille a reçu le prénom "ANNE" depuis 1900.

In []:

```
Nb_total = 0
for prenom in tab_donnees :
    if prenom['preusuel'] == 'ANNE' and prenom['sexe'] == '2' :
        Nb_total = Nb_total + prenom['nombre']

Nb_total
```

Question 7 : Editez un script permettant de connaître le nombre d'enfants nés la même année que vous et qui portent votre prénom.

In []:

Question 8 : Editez un script permettant de connaître le nombre total d'enfants nés depuis 1900 et qui portent votre prénom.

In []:

Question 9 : Editez un script permettant de connaître le prénom le plus donné l'année de votre naissance pour les enfants de votre sexe.

In []:

2.3 – Tri d'un tableau

Pour exploiter les données d'un tableau, notamment pour les tableau de grande taille, il souvent nécessaire de les **trier**. La recherche d'un élément dans un tableau trié est beaucoup efficace que la recherche d'un élément dans un tableau quelconque. Pour cela il faut utiliser la fonction **sorted** ou la méthode **sort**.

La ligne de code suivante trie les données du fichier **nat2018.csv** selon le nombre de fois que le prénom a été donné selon l'ordre croissant.

In []:

```
def cle_nombre(p) :  
    return p['nombre']  
  
tab_donnees.sort(key=cle_nombre)  
tab_donnees
```

La ligne de code suivante trie les données du fichier **nat2018.csv** selon le nombre de fois que le prénom a été donné selon l'ordre décroissant.

In []:

```
def cle_nombre(p) :  
    return p['nombre']  
  
tab_donnees.sort(key=cle_nombre, reverse=True)  
tab_donnees
```

Question 10 : Indiquez quel est le prénom féminin qui a été donné le plus de fois en une seule année.

Il est possible de trier des tableaux selon plusieurs critères. Le code ci-dessous permet de trier les données du fichier **nat2018.csv** selon l'année de naissance puis selon le nombre de fois que le prénom a été donné.

In []:

```
def cle(n) :  
    return (n['annais'], n['nombre'])  
  
tab_donnees.sort(key=cle, reverse=True)  
tab_donnees
```

3 – EXEMPLE DE TRAITEMENT DE TABLEAUX

Tout au long de l'année, la Mairie de Paris assure la surveillance du patrimoine arboré et recense chacun d'eux dans le fichier **arbres.csv** disponible sur son site internet. Le tableau comporte un certain nombre de données telles que :

- l'arrondissement de plantation (ARRONDISSEMENT)
- la classification : nom français (LIBELLEFRANCAIS), genre (GENRE) et l'espèce (ESPECE)
- la circonférence (CIRCONFERENCEENCM)
- la hauteur (HAUTEUR(m))
- les coordonnées géographiques (geo_point_2d)

Question 11 : Editez un script permettant d'importer les données contenues dans le fichier **arbres.csv**.

In []:

Question 12 : Editez un script permettant d'indiquer quelle espèce arbre a la plus grande circonférence et l'arrondissement où il est planté.

In []:

Question 13 : Editez un script permettant d'indiquer quelle espèce arbre a la plus grande hauteur et l'arrondissement où il est planté.

In []:

Certains arbres provoquent de fortes allergies au pollen. Voici la liste de ces arbres :

- Aulne
- Bouleau
- Charme
- Cyprès
- Frêne
- Mûrier
- Olivier
- Platane

Question 14 : Editez un script qui retourne le nombre total d'arbres plantés à Paris pour chacune des 9 espèces précédentes. Ainsi que le nombre total d'arbres provoquant des allergies au pollen. Précisez quelle espèce est la plus présente à Paris.

In []:

Question 15 : Quel arrondissement peut-on conseiller aux personnes souffrant d'allergie au pollen

In []: