

# Manipulation de tables.

---

Une des utilisations principales de l'informatique de nos jours est le traitement de quantités importantes de données dans des domaines très variés. : un site de commerce en ligne peut avoir à

gérer des bases de données pour des dizaines de milliers d'articles en vente, de clients, de commandes , un hôpital doit pouvoir accéder efficacement à tous les détails de traitements de ses patients.....

Il existe des logiciels de traitements de données , et l'étude des bases de données est une matière importante en informatique. Cela est d'ailleurs au programme de terminale.

Pour cette année, nous allons apprendre les opérations de base sur les tables de données en utilisant

Python.

## I. Le format CSV.

---

Le format CSV (pour comma separated values, soit en français valeurs séparées par des virgules) est un format très pratique pour représenter des données structurées.

Dans ce format, chaque ligne représente un enregistrement et sur, une même ligne, les différents champs de l'enregistrement sont séparés par une virgule (d'où le nom). En pratique, on peut spécifier le caractère utilisé pour séparer les différents champs et on utilise fréquemment un point-virgule,

une tabulation ou deux points pour cela.

De plus, la première ligne d'un tel fichier est en général utilisée pour indiquer le nom des différents

champs. Dans ce cas, le premier enregistrement apparaît en deuxième ligne du fichier.

Nous utiliserons dans ce chapitre le fichier `musees_france_2021.csv` que vous avez reçu.

Les données sont issues du site <https://data.gouv.fr> et les données ont été légèrement modifiées.

Question 1 :

1. Ouvrir le document `musees_france_2021` avec le bloc note.
2. Quel est le séparateur de champs utilisés ?
3. Lister le nom des différents champs.
4. Ouvrir le document `musees_france_2021` avec le tableur d'openoffice (ou de libreoffice).  
Veillez à sélectionner convenablement de séparateur. Que constatez vous ?

Dans le bloc note, copier le texte suivant, sauvegardez le au format csv

```
Nom;Prenom;Age;Ville_destination;Region_destination;interet
Dupont;Lucien;25;Tours;37;Château
Fergus;Sophie;35;Marseille;13;Ecomusée
Henocque;Rabah;56;Libourne;33;Chapelle
Masquette;Juliette;27;Lille;59;Exposition
```

Il s'agit des clients d'une agence de voyage. Leur ville et leur destination de voyage est indiqué et l'agence souhaite, en utilisant le document `musees_france_2021` leur fournir une liste de musée qui pourrait les intéresser et à proximité de leur lieu de vacances.

Nous allons apprendre comment répondre à cette question sans faire les choses à la main.

## II. Importation des données.

---

Nous allons utiliser la bibliothèque csv en python.

### 1. Une première solution.

---

```
import csv
musee=[]
documents = open("musees_france_2021.csv",newline="")
lecture = csv.reader(documents,delimiter=";")
for ligne in lecture:
    musee.append(ligne)
documents.close()
```

Question 2 :

1. Copier et exécuter le code ci-dessus.
2. Afficher musee[0] et musee[1]
3. Quel le type de la variable musee ?
4. Que se passe-t-il pour les descriptions des champs ?
5. A quoi sert la fonction reader ?
6. Pourquoi cette méthode ne semble-t-elle pas pertinente ?

### 2. Une meilleure méthode.

---

```
import csv
musee=[]
documents = open("musees_france_2021.csv",newline="")
lecture = csv.DictReader(documents,delimiter=";")
for ligne in lecture:
    musee.append(dict(ligne))
documents.close()
```

Question 3 :

1. Copier et exécuter le code ci-dessus.
2. Afficher musee[0] et musee[1]
3. Quel le type de la variable musee ?
4. Que se passe-t-il pour les descriptions des champs ?
5. A quoi sert la fonction DictReader ?
6. Pourquoi cette méthode ne semble-t-elle pas pertinente ?

Bilan : Afin d'exploiter un fichier csv, il faut importer le module csv, ouvrir le document en veillant à spécifier newline= » » et utiliser la fonction DictReader de manière à ce que chaque entrée soit liée à un champ.

## III. Rechercher une donnée.

---

En utilisant la construction d'une liste par compréhension avec une condition, il est possible, d'obtenir des données précises.

Question 4 :

1. Prenons l'exemple de Juliette qui souhaite avoir le nom et la ville de tous les musées
2. dans le département du 59.
  - a. Entrer dans le script la ligne de code suivante.

```
musee_59=[(m['Nom'],m['ville']) for m in musee if m['Département']=='59']
```
  - b. A l'aide d'une boucle, afficher le contenu de la variable musee\_59.
  - c. Cela répond-il à la question ?
    2. a. Déterminer la liste des noms et des numéros de téléphone des musées de la ville de Marseille.
    - b. Éliminer tous ceux pour lesquels les numéros de téléphone ne sont pas renseignés.
3. a. Déterminer la liste des châteaux dans le département de l'Indre et Loire.
  - b. Lucien, peu satisfait du résultat, souhaite étendre sa recherche aux départements limitrophes à l'Indre et Loire. Lucien aura-t-il de quoi occuper ses vacances ?
4. Déterminer la liste des musées dont la latitude est supérieure à 50.
5. Déterminer la liste des Chapelle et des Ecomusées.
6. Déterminer la liste des musées ayant une latitude supérieure à 50 ou inférieure à 45.

## IV. Tri.

---

Pour exploiter des données, il peut être intéressant de les trier.

Toutefois, si nous reprenons notre exemple, cela ne veut pas dire grand chose de trier le tableau musee. Il faut indiquer selon quels critères on veut effectuer ce tri. Pour cela, on peut soit utiliser la fonction sorted soit la méthode sort avec l'argument supplémentaire qui est une fonction renvoyant la valeur utilisée pour le tri.

Question 5 :

1. a. Entrer dans le script le code suivant.

```
def ville(l):  
    return l['ville']  
musee.sort(key=ville)
```

- b. Afficher les trente premiers noms des musées ainsi que leurs villes de la table musee.
2. Trier la table musee par ordre croissant des longitudes.

Il est également possible de trier en utilisant deux critères. Pour le faire avec une unique instruction, il faut choisir le même ordre (croissant ou décroissant), sinon, il est nécessaire de s'y reprendre en deux fois.

Question 6:

1. Trier la table musee par ordre croissant des latitudes et des longitudes.
2. Trier la table musee par ordre croissant des départements et par ordre décroissant des villes.

## V. Fusion.

---

Il est fréquent lorsqu'on travaille avec des données, et plus particulièrement des données en table, de se retrouver avec plusieurs jeux de données. Une des actions que l'on peut avoir à faire est de combiner les différentes tables pour n'en faire qu'une seule. Cela s'appelle la fusion des tables.

Nous travaillerons, dans cette partie avec deux fichiers:

- le fichier `fichier_client` qui liste les numéros de référence des clients ainsi que leurs noms.

	A	B	C	D	E
1	n_client	nom	prenom	ville	
2	1212	Lacasse	Aubrey	Annecy	
3	1343	Primeau	Angelette	Tours	
4	2454	Gabrieaux	Julie	Bordeaux	
5	895	Gaulin	Élodie	Lyon	
6	2324	Jobin	Dorene	Bourges	
7	34	Boncoeur	Kari	Nantes	
8	1221	Panzeau	Olympia	Metz	
9	1114	Palement	Inès	Bordeaux	
10	3435	Chrétien	Adèle	Moulin	
11	5565	Neufville	Ila	Toulouse	
12	2221	Larivière	Alice	Tours	
13					

- le fichier `fiches_com.csv` qui liste les numéros de commande passées en leur associant la date ainsi que la référence du client qui a passé la commande.

	A	B	C	D
1				
2	n_commande	date	n_client	
3	1111	22/09/18	1343	
4	2323	20/03/19	34	
5	987	12/09/15	5565	
6	454	08/07/14	2324	
7	1567	05/12/18	2221	
8	45	02/02/12	2454	
9	123	04/11/13	5565	
10	2122	12/02/19	3435	
11	1989	04/12/18	1212	
12				

L'objectif est de trouver une méthode permettant de fusionner ces deux tables afin que le numéro de la commande passée soit associé au client.

Pour cela, nous allons utiliser une opération de jointure avec une clé commune qui sera le numéro de client. Il nous faudra également apprendre à écrire dans un fichier csv afin de créer une nouvelle table regroupant toutes ces informations.

Question 7:

- Télécharger les fichiers `fichier_client.csv`, `fiches_com.csv` et `jointure.py` dans le même fichier.
- Ouvrir le fichier `jointure.py` et exécuter la ligne de commande

```
jointures("fiches_client.csv", "fiches_com.csv", "n_client")
```

Qu'effectue cette ligne de commande?

- Commenter la fonction `jointures`.
- Exécuter les lignes de commandes suivantes:

```
nvtab=jointures("fiches_client.csv", "fiches_com.csv", "n_client")  
vers_csv(nvtab, "clients_commandes")
```

- Ouvrir le fichier dans lequel vous avez téléchargé les documents précédents. Que constatez vous?

6. Commenter la fonction `vers_csv`.