

## TP1 : Initiation et prise en main du langage Python

Modules : NumPy, Matplotlib, Pandas, fichiers CSV, etc.

### Objectifs

À la fin de ce TP, l'étudiant sera :

- Manipuler les bases du langage Python (listes, boucles, fonctions)
- Utiliser les bibliothèques **NumPy**, **Pandas**, et **Matplotlib**
- Charger, analyser et visualiser des **données CSV**
- Effectuer des calculs statistiques simples sur des données

**Pré-requis :** Avant de commencer, assurez-vous d'avoir installé :

```
pip install numpy pandas matplotlib
```

### Structure du TP

1. Manipulation des structures de base de Python
2. Introduction à NumPy
3. Analyse de données avec Pandas
4. Visualisation avec Matplotlib
5. Étude de cas sur un fichier CSV réel

#### **1. PARTIE 1 — RÉVISION DES BASES PYTHON**

##### **Exercice 1 : Listes et boucles**

**Énoncé :** Créez une liste contenant les notes de 10 étudiants et affichez :

- la moyenne,
- la note maximale,
- la note minimale.

```
notes = [12, 15, 9, 18, 14, 10, 16, 8, 19, 11]
```

```
moyenne = sum(notes) / len(notes)
print("Moyenne :", moyenne)
print("Max :", max(notes))
print("Min :", min(notes))
```

##### **Exercice 2 : Fonctions et compréhension de liste**

**Énoncé :** Écrire une fonction `carres(liste)` qui renvoie une nouvelle liste contenant le carré de chaque élément.

```
def carres(liste):  
    return [x**2 for x in liste]  
  
print(carres([1, 2, 3, 4, 5]))
```

## **PARTIE 2 — INITIATION À NUMPY**

### **Exercice 3 : Tableaux NumPy**

**Énoncé :** Créez un tableau NumPy de 10 nombres aléatoires entre 0 et 100, et affichez :

- la moyenne,
- l'écart-type,
- le maximum.

```
import numpy as np  
  
data = np.random.randint(0, 100, 10)  
print("Tableau :", data)  
print("Moyenne :", np.mean(data))  
print("Écart-type :", np.std(data))  
print("Maximum :", np.max(data))
```

### **Exercice 4 : Opérations vectorisées**

**Énoncé :** Créez deux vecteurs a et b de taille 5 et calculez :

- leur somme,
- leur produit terme à terme,
- le produit scalaire.

```
a = np.array([1, 2, 3, 4, 5])  
b = np.array([10, 20, 30, 40, 50])  
  
print("Somme :", a + b)  
print("Produit élémentaire :", a * b)  
print("Produit scalaire :", np.dot(a, b))
```

## **PARTIE 3 — ANALYSE DE DONNÉES AVEC PANDAS**

### **Exercice 5 : Création d'un DataFrame**

**Énoncé :** Créez un DataFrame représentant des étudiants avec leurs notes.

```
import pandas as pd  
  
data = {
```

```
"Nom": ["Ali", "Sophie", "Karim", "Lea"],  
"Maths": [12, 18, 14, 10],  
"Physique": [15, 17, 13, 9],  
"Informatique": [16, 19, 15, 12]  
}
```

```
df = pd.DataFrame(data)  
print(df)
```

**Remarque:** Le DataFrame (df) est une structure de données tabulaire fournie par la bibliothèque pandas en Python. Il fonctionne un peu comme un tableau Excel ou une table dans une base de données :

- Les lignes représentent des enregistrements (ici, les étudiants : Ali, Sophie, Karim, Lea).
- Les colonnes représentent des variables (ici : *Nom*, *Maths*, *Physique*, *Informatique*).

Affichez la moyenne des notes par matière et la moyenne générale par étudiant.

```
print("Moyenne par matière :")  
print(df.mean(numeric_only=True))
```

```
df["Moyenne"] = df[["Maths", "Physique", "Informatique"]].mean(axis=1)  
print("\nMoyenne par étudiant :")  
print(df[["Nom", "Moyenne"]])
```

## **PARTIE 4 — MANIPULATION DE FICHIERS CSV**

Pour **créer un fichier CSV** en Python (par exemple à partir d'un DataFrame), tu peux utiliser la bibliothèque **pandas**, qui rend cela très simple.

### **1. À partir d'un DataFrame existant**

En reprenant ton exemple

```
import pandas as pd  
# Données  
data = {  
    "Nom": ["Ali", "Sophie", "Karim", "Lea"],  
    "Maths": [12, 18, 14, 10],  
    "Physique": [15, 17, 13, 9],  
    "Informatique": [16, 19, 15, 12]  
}  
# Création du DataFrame  
df = pd.DataFrame(data)  
# Sauvegarde dans un fichier CSV  
df.to_csv("notes_etudiants.csv", index=False, encoding='utf-8')
```

Cela crée un fichier nommé **notes\_etudiants.csv** dans le dossier courant.

L'option `index=False` évite d'ajouter la colonne d'index automatique (0, 1, 2, 3...).

L'option `encoding='utf-8'` garantit que les accents (é, à, etc.) s'affichent correctement.

## 2. Créer un CSV manuellement (sans pandas)

Tu peux aussi le faire avec le module `csv` de Py

### import csv

#### # Données

```
data = [  
    ["Nom", "Maths", "Physique", "Informatique"],  
    ["Ali", 12, 15, 16],  
    ["Sophie", 18, 17, 19],  
    ["Karim", 14, 13, 15],  
    ["Lea", 10, 9, 12]  
]
```

#### # Écriture dans un fichier CSV

```
with open("notes_etudiants.csv", "w", newline="", encoding="utf-8") as f:  
    writer = csv.writer(f)  
    writer.writerows(data)
```

Résultat identique, mais ici tu n'as pas besoin de pandas.

## Exercice 7 : Lecture d'un fichier CSV

Créez un fichier CSV nommé `etudiants.csv` contenant :

```
Nom,Maths,Physique,Informatique  
Ali,12,15,16  
Sophie,18,17,19  
Karim,14,13,15  
Lea,10,9,12
```

Puis lisez-le avec Pandas :

```
import pandas as pd  
  
df = pd.read_csv("etudiants.csv")  
print(df.head())
```

## ➡ Exercice 8 : Écriture d'un fichier CSV

**Énoncé :** Ajoutez une colonne Moyenne et sauvegardez le DataFrame dans un nouveau fichier.

```
df["Moyenne"] = df[["Maths", "Physique", "Informatique"]].mean(axis=1)
df.to_csv("etudiants_moyennes.csv", index=False)
print("Fichier enregistré ")
```

**Modifier une valeur :**

```
df.loc[df["Nom"] == "Lea", "Maths"] = 11 # Change la note de Lea en maths
```

**Supprimer une colonne :**

```
df = df.drop(columns=["Physique"])
```

## PARTIE 5 — VISUALISATION AVEC MATPLOTLIB

### ☛ Exercice 9 : Graphique simple

**Énoncé :** Tracez la moyenne de chaque étudiant sous forme de graphique.

```
import matplotlib.pyplot as plt

plt.bar(df["Nom"], df["Moyenne"], color="skyblue")
plt.title("Moyenne générale des étudiants")
plt.xlabel("Étudiant")
plt.ylabel("Moyenne")
plt.show()
```

### ☛ Exercice 10 : Courbe et histogramme

**Énoncé :** Tracez un **histogramme** des notes en informatique et une **courbe** pour les notes de physique.

```
plt.figure(figsize=(10,4))

plt.subplot(1,2,1)
plt.hist(df["Informatique"], bins=5, color="orange")
plt.title("Histogramme des notes en Informatique")

plt.subplot(1,2,2)
plt.plot(df["Nom"], df["Physique"], marker="o", color="green")
plt.title("Courbe des notes en Physique")

plt.tight_layout()
plt.show()
```

### 💡 Questions de réflexion



1. Quelle est la différence entre une liste Python et un tableau NumPy ?
2. Pourquoi Pandas est-il plus adapté que NumPy pour les fichiers CSV ?
3. Comment choisir un bon type de graphique selon les données ?

### **Travail à rendre**

- Un fichier **TP1\_python\_donnees.py** contenant les solutions
- Un rapport succinct expliquant :
  - Les bibliothèques utilisées
  - Les étapes suivies
  - Les résultats observés