

COURS COMPLET – SEMAINE 1 : NumPy

1. Installation et préparation (Résumé)

Instructions pour installer Python, créer et activer un environnement virtuel, et configurer VS Code. Vérifier que l'environnement est activé (préfixe (venv) dans le terminal).

Commande essentielle :

```
python -m venv venv
# puis activer:
# Windows (PowerShell): .\venv\Scripts\Activate.ps1
# Git Bash: source venv/Scripts/activate
pip install numpy pandas matplotlib scikit-learn notebook ipykernel
```

Exemple de sortie : test_env.py

```
Python 3.11.5 NumPy 2.1.2 Pandas 2.2.3 test_env.py: 1 2 3 4 5 6
```

2. Rappels Python essentiels

Variables, types (int, float, str, bool), structures (list, tuple, dict, set), conditions, boucles, fonctions et POO. Exemples de code et exercices pratiques fournis.

3. Introduction à NumPy et création de tableaux

Création : np.array, np.zeros, np.ones, np.arange, np.linspace. Propriétés : shape, ndim, size.

```
import numpy as np
a = np.array([1,2,3])
b = np.zeros((2,3))
print(b.shape, b.ndim, b.size)
```

4. Indexation, slicing et opérations mathématiques

Opérations vectorisées, fonctions np.sqrt, np.exp, agrégations np.sum, np.mean, etc.

```
A = np.arange(1,13).reshape(3,4)
print(A[0,1])
print(np.mean(A))
```

Exemple de statistiques et transformation

```
Tableau A : [14 17 12 20 23 15 19 10 30 11 22 28 44 15 16 77 88 99 10 5] Moyenne : 44.57 Minimum : 5 Maximum : 99
Variance : 614.4677 Tableau B (copie de A) : [15 20 8 10 ...]
```

5. Reshape, concaténation et copies

Reshape, flatten, ravel, transpose, vstack, hstack, stack, split. Différence entre view et copy.

```
a = np.arange(12)
b = a.reshape((3,4))
c = b.copy()
```

6. Sauvegarde et chargement (.npy / .csv)

Utiliser `np.save` / `np.load` et `np.savetxt` / `np.loadtxt`. Le format `.npy` est préféré pour les données numériques.

```
Sauvegarde : np.save('data.npy', tab_vendues) Rechargement : A = np.load('data.npy')  
Print : Tous produits vendus : Total  
annuel : 25400 Moyenne mensuelle : 2116.6666666666666 Jan (12500) Fév (15200) Mar (17800)  
Avr (13600) Mai (19300) Juin (22400) Juil (20500) Août (18900) Sept (17600) Oct (21300)  
Nov (23800) Déc (25500) Sommes trimestrielles :  
(14100, 62200, 67600)
```

7. Mini-projet : Analyse statistique des ventes (TP de fin de semaine)

Données mensuelles des ventes (12 mois). Tâches : total annuel, moyenne, mois min/max, reshape en 3x4, sommes trimestrielles, sauvegarde en `.npy` et `.csv`, rechargement et vérification. Ventes :
[12500,15200,17800,13600,19300,22400,20500,18900,17600,21300,23800,25500]

```
import numpy as np  
def get_mois(a):  
    mois = np.array(["Jan", "Fév", "Mar", "Avr", "Mai", "Juin", "Juil", "Août", "Sep", "Oct", "Nov", "Déc"])  
    return mois[a] if 0 <= a < 12 else "Index invalide"  
  
t_vente = np.array([12500,15200,17800,13600,19300,22400,20500,18900,17600,21300,23800,25500])
```

```
print("Total:", t_vente.sum())
print("Moyenne:", t_vente.mean())
idx_min = t_vente.argmin(); idx_max = t_vente.argmax()
print("Min:", t_vente[idx_min], get_mois(idx_min))
print("Max:", t_vente[idx_max], get_mois(idx_max))
t_res = t_vente.reshape(3,4)
print("Sommes trimestrielles:", t_res.sum(axis=1))
np.save("tab_original.npy", t_vente)
np.savetxt("tab_original.csv", t_vente, fmt="%d", delimiter=",")
```

Glossaire NumPy (sélection)

- ndarray: structure principale - shape: dimensions - dtype: type de données - ravel/flatten: aplatir - dot/@: produit matriciel

Exercices supplémentaires

1) Implémenter une fonction qui normalise un tableau (min-max). 2) Charger un dataset CSV simple et calculer corrélation entre colonnes. 3) Implémenter un split train/test aléatoire.

Document généré pour la Semaine 1 du programme IA. Contient code, exemples et sorties simulées.