

# TP NUMPY v2



## OBJECTIFS

Les objectifs pédagogiques de ce TP sur NumPy visent à fournir aux apprenants les compétences et connaissances nécessaires pour manipuler et analyser des données numériques en Python à l'aide de la bibliothèque NumPy. Voici les principaux objectifs pédagogiques. Dans votre compte rendu vous devrez en quelques phrases préciser ce que chaque objectif évoque pour vous. Votre évaluation porte sur la réponse mais aussi sur la présentation de cette dernière : **les illustrations, explications, documentations détaillées et enrichissement** seront particulièrement valorisés dans la notation.

1. Comprendre les Fondamentaux de NumPy
2. Maîtriser la Création et Manipulation de Tableaux
3. Appliquer des Opérations Mathématiques et Statistiques
4. Explorer les Capacités Algébriques Linéaires de NumPy
5. Utiliser NumPy pour le Traitement d'Image
6. Développer des Compétences en Analyse de Données
7. Résoudre des Problèmes Réels
8. Préparation à l'Apprentissage de Bibliothèques Avancées

En atteignant ces objectifs, les apprenants seront bien équipés pour utiliser NumPy efficacement dans leurs projets de programmation Python, renforçant ainsi leur capacité à travailler avec des données complexes et à mener des analyses numériques sophistiquées.

## PRÉALABLE

Réalisez et documentez ce tutoriel :

<https://realpython.com/numpy-tutorial/>

Visionner puis résumer la vidéo :

<https://www.youtube.com/watch?v=vw4u9uBFFqU>

Produire les codes de la vidéo et les commenter

## QUESTIONS

Question 1: Qu'est-ce que NumPy et pourquoi est-il important en science des données ?

Question 2: Comment installer NumPy ?

Question 3: Comment importer NumPy dans un projet Python ?

Question 4: Créez un tableau NumPy de taille 5 rempli de zéros.

Question 5: Comment générer un tableau de nombres entiers de 10 à 50 ?

Question 6: Créez une matrice identité de taille 4x4.

Question 7: Utilisez NumPy pour générer un nombre aléatoire entre 0 et 1.

Question 8: Générez un tableau de 25 nombres aléatoires tirés d'une distribution normale standard.

Question 9: Créez la matrice suivante à l'aide de NumPy :

Question 10: Sélectionnez l'élément "6" de la matrice ci-dessus.

Question 11: Produisez un tableau de 20 points linéairement espacés entre 0 et 1.

Question 12: Multipliez tous les éléments de la matrice suivante par 2.

Question 13: Comment inverser les lignes et les colonnes d'une matrice 2D ?

Question 14: Comment accéder aux éléments d'un tableau multidimensionnel ?

Question 15: Comment calculer la somme de tous les éléments d'un tableau NumPy ?

Question 16: Comment calculer la moyenne des éléments d'un tableau NumPy ?

Question 17: Comment créer un tableau 2D avec des valeurs aléatoires ?

Question 18: Comment remplacer les éléments d'un tableau NumPy qui sont supérieurs à un certain seuil par 0 ?

Question 19: Comment trouver les indices des éléments non nuls dans un tableau NumPy ?

Question 20: Comment effectuer une opération de multiplication matricielle entre deux tableaux NumPy ?

Question 21: Créez une matrice 5x5 avec des valeurs 1,2,3,4 juste en dessous de la diagonale.

Question 22: Calculez le déterminant d'une matrice 3x3.

Question 23: Inverser une matrice 3x3.

Question 24: Trouvez les valeurs et vecteurs propres d'une matrice 3x3.

## POUR ALLER PLUS LOIN

Question 25: Utilisez la "fancy indexing" pour extraire des éléments non contigus d'un tableau (par exemple, extraire les éléments aux positions [1, 3, 8]).

Question 26: Créez un tableau 2D de formes (5,5) rempli de valeurs aléatoires et normalisez-le (soustrayez la moyenne et divisez par l'écart type).

Question 27: Comment calculer la distance euclidienne entre deux tableaux NumPy A et B sans utiliser de boucle ?

Question 28: Comment trouver les éléments communs entre deux tableaux NumPy ?

Question 29: Créez un tableau 2D représentant un plan d'échiquier (8x8), où les cases alternent entre 0 (blanc) et 1 (noir).

Question 30: Comment appliquer une fonction personnalisée à chaque élément d'un tableau NumPy ?

Question 31 : trouvez 3 exemples d'utilisation de numpy dans la résolution de 3 problèmes de traitement de données différents. Expliquez à votre binôme ou votre voisin en quelques minutes deux résolutions sur 3.

Ces questions avancées englobent une gamme de techniques et de fonctions NumPy, des opérations linéaires et algébriques complexes aux manipulations de tableaux sophistiquées. Maîtriser ces concepts avancés est crucial pour la résolution de problèmes complexes en science des données, en analyse numérique et en ingénierie logicielle.