

# Progammation Scientifique: Numpy

Master 1 , Data Science & IA

Ibrahima SY

11/18/2021

## **Problème 1 :**

Considérons le vecteur suivant :

$$x = [1, 2, 3, 4, 5]$$

1. Créer ce vecteur à l'aide d'un tableau que l'on appellera **x**.
2. Afficher le type de **x** puis sa longueur.
3. Extraire le premier élément, puis en faire de même avec le dernier.
4. Extraire les trois premiers éléments et les stocker dans un vecteur que l'on nommera **a**.
5. Extraire les 1er, 2e et 5e éléments du vecteur (attention aux positions) ; les stocker dans un vecteur que l'on nommera **b**.
6. Additionner le nombre 10 au vecteur **x**, puis multiplier le résultat par 2.
7. Effectuer l'addition de **a** et **b**, commenter le résultat.
8. Effectuer l'addition suivante : **x+a** ; commenter le résultat, puis regarder le résultat de **a+x**.
9. Multiplier le vecteur par le scalaire **c** que l'on fixera à 2.
10. Effectuer la multiplication de **a** et **b** ; commenter le résultat.
11. Effectier la multiplication suivante : **x\*a** ; commenter le résultats.
12. Récupérer les positions des multiples de 2 et les stocker dans un vecteur que l'on nommera **ind**, puis conserver uniquement les multiples de 2 de **x** dans un vecteur que l'on nommera **mult\_2**.
13. Afficher les éléments de **x** qui sont multiples de 3 et multiples de 2.
14. Afficher les éléments de **x** qui sont multiples de 3 ou multiples de 2.
15. Calculer la somme des éléments de **x**.
16. Remplacer le premier élément de **x** par un 4.
17. Remplacer le premier élément de **x** par la valeur **NaN**, puis calculer la somme des éléments de **x**.
18. Supprimer le vecteur **x**

## **Problème 2 :**

Créer la matrice :

$$A = \begin{pmatrix} -3 & 5 & 6 \\ -1 & 2 & 2 \\ 1 & -1 & -1 \end{pmatrix}$$

1. Afficher la dimension de **A**, son nombre de colonnes, son nombre de lignes et sa longueur.

2. Extraire la seconde colonne de **A**, puis la première ligne. 4.Extraire l'élément en troisième position à la première ligne.
3. Extraire la sous-matrice de dimension  $2 \times 2$  du coin inférieur de **A**, c'est-à-dire:

$$A = \begin{pmatrix} 2 & 2 \\ -1 & -1 \end{pmatrix}$$

4. Calculer la somme des colonnes puis des lignes de **A**.
5. Afficher la diagonale de **A**.
6. Rajouter le vecteur  $(1 \ 2 \ 3)^T$  à droite de la matrice **A** et stocker le résultat dans un objet appelé **B**.
7. Retirer le quatrième vecteur de **B**.
8. Retirer la première et la troisième ligne de **B**.
9. Ajouter le scalaire 10 à **A**.
10. Ajouter le vecteur  $(1 \ 2 \ 3)^T$
11. Ajouter la matrice identité  $I_3$  à **A**.
12. Diviser tous les éléments de la matrice **A** par 2.
13. Multiplier la matrice **A** par le vecteur  $(1 \ 2 \ 3)^T$
14. Afficher la transposée de **A**.
15. 1. Effectuer le produit avec transposition  $A^T A$

### **Problème 3 : calcul du temps d'exécution de l'élévation d'un tableau à la puissance 2**

Vous allez prendre un tableau **a** des entiers de 1 à 10000 et les élever à la puissance 2 de plusieurs manière en calculant le temps d'exécution. Pour calculer  $e^2$ , vous pouvez essayer: `e*e`, `e**2`, `np.power(e, 2)` ou `np.square(e)`:

1. Vous créer une liste Python vide. Vous itérez sur les éléments du tableau **a**, avec un `for-Python`, en ajoutant chaque élément élevé à la puissance 2 dans la liste.
2. Vous créez un `np.ndarray`, non initialisé, pour stocker les éléments. Vous itérez sur les éléments du tableau **a** avec un `for-Python`, en ajoutant chaque élément élevé à la puissance 2 (`x**2`) dans le `np.ndarray`
3. Vous construisez la liste Python, initialisée avec les puissances 2 des éléments de **a**, avec une compréhension Python.
4. Vous le faites de manière vectorisée en utilisant l'opérateur de puissance (`**2`).
5. Vous le faites de manière vectorisée en utilisant l'opération de puissance (`np.power`).
6. Quelles sont les manières de faire les plus rapides ?