Introduction au Big Data – TP3 – Numpy

Dans ce TP nous allons utiliser les fonctions de Numpy pour construire des tableaux et les transformer. Faites ce TP sous Visual Studio Code (dans la fenêtre Interactive, avec l'invite de la Debug Console, ou bien dans un programme .py ou Jupyter). Vous pouvez aussi utiliser n'importe quel interpréteur Python avec Numpy installé.

Vous pouvez utiliser autant d'instructions Numpy que nécessaire, il n'est pas question d'écrire des expressions en une seule ligne (ce qui est rarement faisable).

L'essentiel est de ne pas utiliser de programmation impérative (boucles, conditions…).

1. Ecrire du code Numpy permettant de passer du premier au second tableau

   np.array([1 , 2 , 3 , 4 , 5])
   np.array([5 , 4 , 3 , 2 , 1])

2. Ecrire du code Numpy permettant de passer du premier au second tableau

   np.array([1 , 2 , 3 , 4 , 5])
   np.array([10 , 30 , 50])

3. Ecrire du code Numpy permettant de passer du premier au second tableau

   np.array([1 , 2 , 3 , 4 , 5])
   np.array([1 , 9 , 25])

4. Ecrire du code Numpy permettant de transformer ce tableau en tableau à 3 rangées avec des valeurs triés de haut en bas et de gauche à droite

   np.array([1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9])

5. Ecrire du code Numpy permettant à partir de ce tableau

   c = np.array([[1,2,3],[4,5,6]])

   d'afficher

   np.array([2, 5])

6. Ecrire du code Numpy qui génère un tableau de cette forme, paramétré par n où n=9

   [[1 2 3 10 20 30]
    [4 5 6 40 50 60]
    [7 8 9 70 80 90]]

7. Ecrire du code Numpy qui génère le tableau suivant, paramétré par n où n=9

   [[1. 1. 1. 1. 1. 1. 1. 1. 1.]

```
[1. 2. 2. 2. 2. 2. 2. 2. 1.]
[1. 2. 2. 2. 2. 2. 2. 2. 1.]
[1. 2. 2. 2. 2. 2. 2. 2. 1.]
[1. 2. 2. 2. 2. 2. 2. 2. 1.]
[1. 2. 2. 2. 2. 2. 2. 2. 1.]
[1. 2. 2. 2. 2. 2. 2. 2. 1.]
[1. 2. 2. 2. 2. 2. 2. 2. 1.]
[1. 1. 1. 1. 1. 1. 1. 1. 1.]]
```

8. Ecrire du code Numpy qui permet de passer de cette valeur

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

à celle-ci

```
[[1 4 6]
 [4 7 9]
 [7 10 12]]
```

9. Considérons les valeurs Numpy suivantes :

```
a = np.array([
  [1, 2, 3, 4, 5],
  [6,7,8,9,10],
  [11,12,13,14,15]])
b = np.array([[0,1], [0,3], [1,4]])
```

comment produire la valeur suivante ?

```
array([[[ 1,  2],
        [ 6,  7],
        [11, 12]],
       [[ 1,  4],
        [ 6,  9],
        [11, 14]],
       [[ 2,  5],
        [ 7, 10],
        [12, 15]]])
```

10. Ecrire du code Numpy qui permet de remplacer les valeurs d'un tableau plus grandes que 10, en mettant pour toutes ces valeurs 20 à la place.

11. Ecrire du code Numpy qui permet de déterminer si un tableau de forme arbitraire ne contient aucune valeur nulle.

12. Ecrire du code Numpy qui permet de déterminer si un tableau de forme arbitraire ne contient que des valeurs nulles.

13. Ecrire du code Numpy qui permet de passer de cette valeur

    [1 2 3 4]

    à celle-ci

    [1 1 2 2 3 3 4 4]

14. Ecrire du Numpy permet, à partir d'un tableau de cette forme

    ```
    array([[ 0,  10,  20,  30],
           [ 40,  50,  60,  70],
           [ 80,  90, 100, 110]])
    ```

    d'extraire les valeurs

    array([[60, 40, 50],[10, 8, 9]])

15. Ecrire du code Numpy qui permet de passer de ce tableau

    ```
    [[1 2 3]
     [4 5 6]
     [7 8 9]]
    ```

    à ce tableau

    ```
    [[1 1 2 2 3 3]
     [1 1 2 2 3 3]
     [4 4 5 5 6 6]
     [4 4 5 5 6 6]
     [7 7 8 8 9 9]
     [7 7 8 8 9 9]]
    ```

16. Avec np.unique (et return_index) et np.split, passer du tableau suivant

    ```
    array([[  1, 1275],
           [  1, 1441],
           [  1, 1494],
           [  1, 1593],
           [  2, 1679],
           [  2, 1533],
           [  2, 1686],
           [  3, 1559],
           [  3, 1219],
           [  3, 1455],
    ```

[4, 1605],
[4, 1468],
[4, 1692],
[4, 1613]])

Au tableau suivant

array([[[1275, 1441, 1494, 1593]],
[[1679, 1533, 1686]],
[[1559, 1219, 1455]],
[[1605, 1468, 1692, 1613]]])

La première colonne de la donnée d'entrée est supposée déjà triée.

17. Même question que la précédente, mais en sachant que les données d'entrées ne sont pas forcément triées suivant la première colonne.