

TP2 –Introduction aux librairies Os, NumPy et Matplotlib

Partie 1 : Interaction avec le système d'exploitation sous Python (module `os`)

Introduction au module `os`

1. Utilisez les commandes du package `os` pour :
 - a. Afficher le répertoire courant ;
 - b. Afficher la liste des répertoires et fichiers dans le répertoire courant ;
 - c. Se placer dans le répertoire *Data* et afficher la liste des fichiers dans le répertoire *Data*.

Partie 2 : Introduction à NumPy

Les bases de NumPy

Liste Python (Python list) vs tableau NumPy (NumPy array)

2. Créer une liste Python $x = [1,4,2,7]$ et un tableau NumPy y contenant les mêmes valeurs. Vérifiez le type de chacune de ces deux variables.
3. Comparez les résultats $x + x$, $y + y$, $2 * x$ et $2 * y$. Pourquoi les résultats sont-ils différents ?

Création des arrays NumPy

4. Utilisez les fonctions `arange` et `linspace` pour créer l'array suivant : $[0, 2, 4, 6]$.
5. Créez deux tableaux $m0$ et $m1$ contenant trois lignes et deux colonnes dont les composantes sont initialisées à 0 et 1, respectivement.

Manipulation des arrays NumPy

6. Utilisez le tableau $y = [1,4,2,7]$ 1D créé précédemment pour répondre aux questions suivantes :
 - a. Affichez le premier élément de y .
 - b. Affichez les trois premiers éléments de y .
 - c. Affichez le dernier élément de y .
 - d. Affichez les deux derniers éléments de y .
 - e. A quoi servent les fonctions `np.size`, `np.shape` et `len` ?
7. Utilisez le tableau $m = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$, pour répondre aux questions suivantes :
 - f. Affichez la composante m_{12} de m .
 - g. Affichez la première ligne de m .
 - h. Affichez un array avec les composantes $[m_{12}, m_{13}]$.
 - i. Comparez les fonctions `np.size`, `np.shape` et `len`.
 - j. Affichez la transposée de m .

Logical indexing

8. A partir des fonctions `list` et `range`, créez une liste de listes $x = [liste1, liste2, liste3]$ composée des trois listes suivantes :
 - k. $liste1 = [0, 1, 2, 3, 4, 5]$
 - l. $liste2 = [9, 10, 11, 12, 13, 14]$
 - m. $liste3 = [20, 21, 22, 23, 24, 25]$
9. Créez l'array NumPy xnp correspondant à cette liste.
10. Que retourne la commande $xnp > 10$?
11. Affichez un array contenant toutes les valeurs de xnp supérieures à 10.

Utilisation de NumPy sur des données

12. Ouvrez un des fichiers de données (fichiers *.atf*) avec un éditeur de texte. A partir de l'en-tête du fichier, répondez aux questions suivantes :
 - a. Quel est le nombre de données ?
 - b. Quel est le pas d'échantillonnage en temps ?
 - c. Quelle est l'unité de temps ?
13. Chargez le premier fichier à l'aide de la fonction `np.loadtxt` dans un array nommé `amp1`. N'oubliez pas de supprimer les en-têtes lors du chargement. La taille du tableau (array) est-elle cohérente avec le nombre d'échantillons ? Chargez les autres fichiers ; nommez-les `amp2`, `amp3`, `amp4`.
14. A partir des données temporelles indiquées dans l'en-tête, créez le vecteur temps correspondant au temps d'enregistrement des données (la première valeur d'amplitude a été enregistrée à $t=0$).
15. Affichez les dix premières valeurs et la taille du vecteur temps pour vérifier que le vecteur temps est correct.

Partie 4 : Introduction à Matplotlib

Exemple basique

16. Créez un vecteur x allant de 0 à 4π et constitué de 200 points.
17. Représentez $\sin(x)$ en fonction de x . A quoi sert la fonction `plt.show()` ?
18. Améliorez la figure :
 - a. Utilisez les fonctions `plt.xlabel()` et `plt.ylabel()` pour ajouter des noms aux axes (augmentez la taille de police (fontsize) à 18).
 - b. Utilisez les fonctions `plt.xticks()` et `plt.yticks()` pour augmenter la taille de police des graduations à 16.
19. Représentez plusieurs courbes sur une même figure:
 - a. Représentez $\sin(x)$ et $\cos(x)$ en fonction de x sur le même graphique.
 - b. Améliorez la mise en forme : augmentez les tailles de police des graduations, ajoutez des titres aux axes et augmentez les tailles de police, ajoutez une légende, limitez l'axe x à $[0,6]$.
20. Créez des subplots (plusieurs graphiques par figure):
 - a. Représentez $\sin(x)$ (en rouge) et $\cos(x)$ (en bleu) en fonction de x sur deux subplots alignés (utilisez la fonction `plt.subplot()`).
 - b. Améliorez la mise en forme des figures (titre des axes, taille de police). Utilisez la fonction `plt.tight_layout()` pour améliorer la disposition des subplots.
21. Représentez $\sin(x)$ en fonction de x de manière orientée objet.

Exemple sur les données

22. Représentez les données sur 4 subplots différents.
23. A quel temps le signal émetteur a-t-il été déclenché ?
24. Tronquez le vecteur temps et les vecteurs des données pour supprimer les données avant le déclenchement du signal.
25. Modifiez le vecteur t de manière à ce que $t = 0$ corresponde au déclenchement du signal émetteur.
26. Représentez les nouveaux signaux. A partir des figures, donnez le temps approximatif de première arrivée de l'onde P.

Pour aller plus loin

27. Créez un tableau m dont la première ligne est $[1,2,3]$ et la deuxième ligne est $[4,5,6]$. Utilisez deux options différentes :
 - Utilisez la fonction `array`.
 - Utilisez la fonction `arange` puis la méthode `reshape`.
28. Ecrire une boucle `for` pour charger les données de manière itérative.
29. Use the `%matplotlib` notebook magic command to pick the first break arrival in a more accurate way.