

Atelier_5

N'oubliez pas d'utiliser des commentaires pour expliquer le fonctionnement de chaque partie de votre code.

Activité 1

Générez un tableau NumPy appelé `donnees` contenant 100 nombres aléatoires entiers entre 1 et 100.

Calculez et affichez la moyenne des valeurs dans le tableau `donnees`.

Calculez et affichez la médiane des valeurs dans le tableau `donnees`.

Calculez et affichez l'écart-type des valeurs dans le tableau `donnees`.

Créez un tableau appelé `donnees_pairs` qui contient uniquement les valeurs paires de `donnees`.

Créez un tableau appelé `donnees_impaires` qui contient uniquement les valeurs impaires de `donnees`.

Trouvez et affichez la valeur maximale et sa position (indice) dans le tableau `donnees`.

Triez le tableau `donnees` par ordre croissant et affichez le résultat.

Vous pouvez utiliser les fonctions: `numpy.random.randint` pour générer des nombres aléatoires entiers. et les fonctions statistiques telles que `mean`, `median`, `std`, et les méthodes comme `max` et `argsort`

Activité 2

Générez deux tableaux NumPy :

`notes_maths` contenant 20 notes aléatoires entre 10 et 20.

`notes_physique` contenant 20 notes aléatoires entre 12 et 18.

Calculez la moyenne, l'écart-type et la médiane de chaque ensemble de notes. Affichez les résultats.

Créez un tableau combiné `notes_combined` en concaténant les deux tableaux précédents.

Trouvez et affichez les indices des éléments dans le tableau combiné qui sont supérieurs à 15.

Remplacez toutes les notes inférieures à 14 dans le tableau combiné par la valeur 14.

Créez un tableau booléen `mentions` qui indique si chaque note dans le tableau combiné est supérieure ou égale à 16.

Utilisez le tableau mentions pour créer un nouveau tableau `etudiants_meritants` contenant les notes des étudiants ayant une mention (supérieure ou égale à 16).

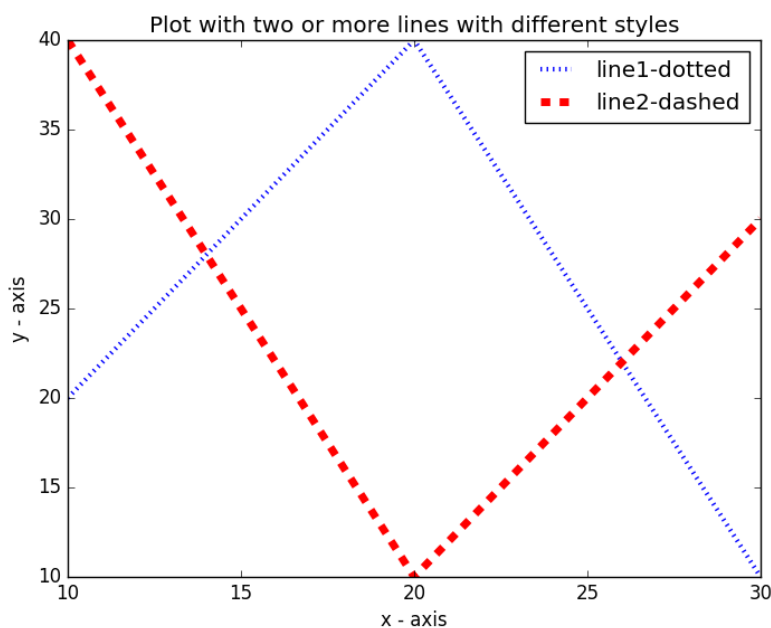
Affichez le pourcentage d'étudiants ayant une mention.

Enregistrez le tableau `notes_combined` dans un fichier CSV avec deux colonnes : "Maths" et "Physique".

Activité 3

Écrivez un programme Python pour tracer deux lignes ou plus avec des styles différents.

L'extrait de code donne la sortie montrée dans la capture d'écran suivante :



Activité 4 (tkinter)

Créer une application graphique avec Tkinter qui affiche un bouton et un compteur. Chaque fois que l'utilisateur clique sur le bouton, le compteur est incrémenté et la nouvelle valeur est affichée.

Instructions :

Créez une fenêtre Tkinter avec un titre approprié.

Ajoutez un label affichant le compteur initialisé à 0.

Ajoutez un bouton avec le texte "Cliquez ici".

Définissez une fonction qui sera appelée à chaque clic sur le bouton. Cette fonction doit incrémenter le compteur et mettre à jour le texte du label.

Lancez la boucle principale de l'application.

Activité 5 (POO et tkinter)

Projet: Carnet d'Adresses

Créez une application de carnet d'adresses simple en utilisant Tkinter et les classes en Python.

L'application doit permettre à l'utilisateur d'ajouter des contacts avec leurs noms et numéros de téléphone, et d'afficher la liste complète des contacts.

Définissez une classe Contact qui a des attributs pour le nom et le numéro de téléphone.

Créez une classe CarnetAdresses qui aura une liste de contacts et des méthodes pour ajouter un contact et récupérer la liste complète des contacts.

Utilisez Tkinter pour créer une interface utilisateur avec les éléments suivants :

Deux champs de saisie pour le nom et le numéro de téléphone.

Un bouton "Ajouter" pour ajouter le contact à la liste.

Une zone de texte pour afficher la liste des contacts.

Lorsque l'utilisateur clique sur le bouton "Ajouter", créez un objet Contact avec les informations saisies, ajoutez-le à la liste du carnet d'adresses, et mettez à jour la zone de texte avec la liste mise à jour.

Affichez le nom complet et le numéro de téléphone de chaque contact dans la zone de texte, chaque contact sur une nouvelle ligne.

Bon travail