Atelier 6

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

Activité Pandas

Considérez un ensemble de données fictif représentant les ventes d'une entreprise. L'ensemble de données est stocké dans un fichier CSV nommé "ventes.csv" avec les colonnes suivantes :

Date: La date de la vente (format: AAAA-MM-JJ).

Produit : Le nom du produit vendu.

Quantité : La quantité vendue.

Prix unitaire: Le prix unitaire du produit.

Tâches à effectuer :

Chargez l'ensemble de données dans une structure DataFrame.

- 1. Affichez les 5 premières lignes du DataFrame.
- 2. Calculez le chiffre d'affaires total réalisé par l'entreprise avec sum()
- 3. Identifiez le produit le plus vendu. Utiliser groupby() et idxmax()
- 4. Trouvez la date à laquelle l'entreprise a réalisé le meilleur chiffre d'affaires journalier.
- 5. Tracez un graphique barres pour visualiser les ventes par produit. Utiliser le paramètre kind avec la fonction plot(kind='bar')

Activité test unitaire :

Test unitaire d'une fonction de calcul

Considérez la fonction suivante qui effectue une opération de multiplication :

calc.py

def multiply(a, b):

return a * b

Écrire des tests unitaires :

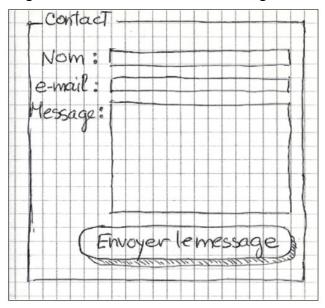
Créez un fichier de test (par exemple, test calc.py) pour tester la fonction multiply.

Écrivez des tests pour vérifier le bon fonctionnement de la fonction dans différentes situations (positif, négatif, zéro, etc.).

Activité Flask

Réaliser ce formulaire suivant dans une template projet flask

Le message écrit doit être affiche dans /message avec le nom et le mail



templates/index.html:

```
<form action="/message" method="post">
 <1i>
   <label for="name">Nom&nbsp;:</label>
   <input type="text" id="name" name="user name" />
  <1i>
   <label for="mail">E-mail&nbsp;:</label>
  <input type="email" id="mail" name="user mail" />
  <1i>
   <label for="msg">Message&nbsp;:</label>
   <textarea id="msg" name="user message"></textarea>
  <button type="submit">Envoyer</button>
```

</form>

templates/message.html:

```
<h1>Message reçu</h1>
Nom : {{ user_name }}
E-mail : {{ user_mail }}
Message : {{ user_message }}
```

Activité Json

Créer un script Python qui utilise des fichiers JSON pour stocker des données.

Ajoutez une fonction create_json_file qui prend un nom de fichier et des données en paramètres, puis écrit ces données dans un fichier JSON.

Lecture d'un fichier JSON:

Ajoutez une fonction read_json_file qui prend un nom de fichier en paramètre et lit les données depuis un fichier JSON.

Modification d'un fichier JSON:

Ajoutez une fonction update_json_file qui prend un nom de fichier, une clé et une nouvelle valeur en paramètres. La fonction devrait mettre à jour la valeur associée à la clé dans le fichier JSON.

Affichage des données :

Ajoutez une fonction display_data qui prend un nom de fichier en paramètre et affiche le contenu du fichier JSON.

Activité: Gestion de fichiers avec subprocess

Créez une fonction Python appelée create_file qui prend un nom de fichier en paramètre et utilise la commande shell touch pour créer ce fichier.

Liste de fichiers:

Créez une fonction Python appelée list_files qui utilise la commande shell ls pour lister les fichiers dans le répertoire courant. La fonction devrait renvoyer la liste des noms de fichiers.

Suppression de fichiers :

Créez une fonction Python appelée delete_file qui prend un nom de fichier en paramètre et utilise la commande shell rm pour supprimer ce fichier.

Modification de fichiers:

Créez une fonction Python appelée append_to_file qui prend un nom de fichier et une chaîne de texte en paramètres, puis utilise la commande shell echo pour ajouter cette chaîne à la fin du fichier.

Bon travail