

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°10 : La gestion des fichiers

BTS CIEL
2 ^{ème} année
Page 1 sur 2

Exercice 1 : Identification des fichiers .txt

Écrivez un programme Python qui explore le répertoire "mesDossier" et identifie tous les fichiers ayant l'extension .txt.

Le programme doit :

1. Gérer les erreurs éventuelles lors de l'accès au répertoire, comme un répertoire inexistant ou des permissions insuffisantes, et afficher un message d'erreur approprié.
2. Filtrer la liste des fichiers pour ne conserver que ceux qui se terminent par l'extension .txt.
3. Afficher les noms de ces fichiers .txt. Si aucun fichier .txt n'est trouvé, afficher un message indiquant que le répertoire ne contient aucun fichier de ce type.

Résultats attendus :

```
Fichiers .txt dans le répertoire :
destination.txt
nouveau_nom.txt
source.txt
```

Exercice 2 : Ajout de préfixes aux fichiers d'un répertoire

Écrivez un programme Python qui ajoute un préfixe spécifié à tous les fichiers d'un répertoire donné (par défaut, le répertoire courant).

Le programme doit :

1. Vérifier si chaque élément listé est bien un fichier avant de tenter de le renommer.
2. Ajouter un préfixe défini par l'utilisateur (dans le code) au nom de chaque fichier, tout en conservant son extension.
3. Renommer chaque fichier en conséquence et afficher une confirmation pour chaque fichier renommé.
4. Gérer les erreurs éventuelles, comme un problème d'accès ou une tentative de renommer un élément qui n'est pas un fichier (par exemple, un dossier), et afficher des messages d'erreur appropriés.

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°5 : Les fonctions

BTS CIEL
2 ^{ème} année
Page 2 sur 2

Exercice 3: Classification des fichiers et dossiers dans un répertoire

Écrivez un programme Python qui parcourt un répertoire donné (par défaut, le répertoire courant) et sépare les fichiers des dossiers.

Le programme doit :

1. Vérifier pour chaque élément s'il s'agit d'un fichier ou d'un dossier.
2. Ajouter les fichiers et les dossiers à des listes distinctes.
3. Afficher séparément les fichiers et les dossiers trouvés dans le répertoire.
4. Si aucun fichier ou dossier n'est trouvé, afficher un message indiquant qu'aucun élément correspondant n'est présent.

Exercice 4 : Copie sélective de fichiers .txt vers un dossier cible

Réalisez un programme Python qui copie tous les fichiers ayant l'extension .txt d'un dossier source vers un dossier de destination.

Le programme doit :

1. Vérifier si le dossier de destination existe ; s'il n'existe pas, le créer automatiquement.
2. Parcourir le contenu du dossier source pour identifier les fichiers avec l'extension .txt.
3. Copier chaque fichier .txt trouvé vers le dossier de destination, tout en conservant les métadonnées des fichiers (comme les dates de modification et de création).
4. Gérer les erreurs potentielles, comme l'impossibilité de lire le dossier source ou de copier un fichier, et afficher des messages d'erreur clairs dans ces cas.

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°5 : Les fonctions

BTS CIEL
2 ^{ème} année
Page 2 sur 2

Exercice 5 : Analyse des Journaux d'Authentification

Vous êtes chargé de développer un script Python capable d'analyser les tentatives d'accès échouées à un serveur Linux en explorant les journaux d'authentification. Le script doit être capable de :

1. Lire le contenu du fichier /var/log/auth.log pour analyser les informations d'accès.
2. Identifier les adresses IP ayant effectué des tentatives échouées.
3. Recenser les utilisateurs ciblés par ces tentatives.
4. Présenter un résumé des données collectées, incluant :
 - Les adresses IP les plus fréquentes.
 - Les utilisateurs les plus ciblés.

Consignes supplémentaires :

- Le fichier à traiter est spécifiquement /var/log/auth.log. Vous devez structurer le script en utilisant des fonctions bien définies pour assurer une meilleure lisibilité et modularité.

Par exemple :

- Une fonction pour lire le contenu du fichier.
 - Une fonction pour analyser les données du fichier.
 - Une fonction pour afficher les résultats de manière claire.
- Si le fichier est introuvable ou inaccessible (par exemple, faute de permissions), la fonction correspondante doit afficher un message d'erreur clair, sans provoquer d'arrêt brutal du script.
- Le script doit être robuste et capable de gérer des cas imprévus, tels que des lignes de logs mal formatées, en les ignorant ou en affichant un avertissement approprié.