



SAE 1.05

Traiter des données

Berge Léandro Le Guilly Tanguy

Sommaire

- Introduction
- Fonctionnement du programme
- Problématiques rencontrées et les solutions
- Conclusion

Introduction

Objectif :

Suppression et localisation de gros fichiers. Cela peut être utile pour les administrateurs sys. (Gagner du temps, et analyser une arborescence)

Compétences :

Manipuler du PowerShell et du python

Fonctionnement

- Creation_Boutons.py
- Creation_Camembert.py
- Creation_Legendes.py
- Creation_Onglets.py
- gros_fichiers.json
- main.py
- powershell_script.ps1
- script1.py
- script2.py
- script3.py

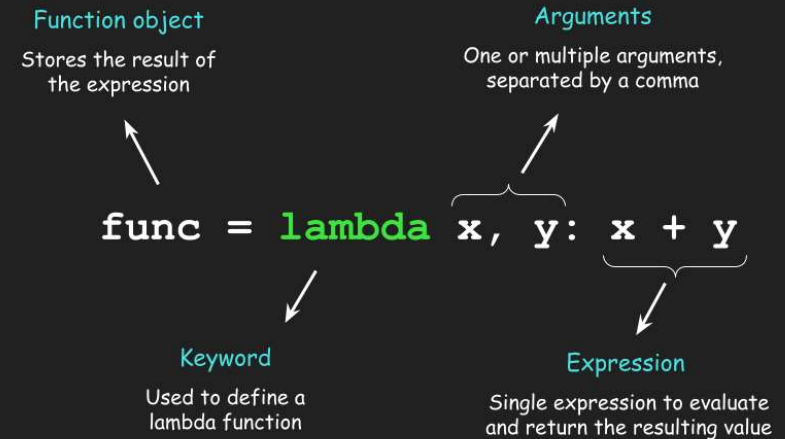
- 1-Lancement script Powershell
- 2-Sélection du répertoire (script1.py)
- 3-Analyse des fichiers et création du JSON (script2.py)
- 4-Affichage du graphe (script3.py)
- 5- Génération du script Powershell de suppr (script3.py)
- 6-Exécution du script Powershell de suppr

Problématiques rencontrées

- L'utilisation de la fonction « Lambda » (La syntaxe) :

```
def trie(listtout):  
    listtout.sort(key=lambda x: x[1], reverse=True)  
    for i in listtout:  
        print(i)  
    return listtout
```

(Script2.py)



Problématiques rencontrées

- La gestion d'erreurs :

```
if not directory:  
    print("Aucun répertoire sélectionné, fin du programme.")  
    exit()
```

(script2.py)

```
if not liste_fichiers:  
    print("Le fichier JSON est vide ou n'existe pas.")  
    sys.exit(1)
```

(script3.py)

```
else:  
    print("Aucun fichier sélectionné pour la suppression.")
```

(script3.py)

(Fonction: callback_suppression())

Problématiques rencontrées

Complexité en temps du programme :

- Script 3 complexité en temps globalement linéaire.

```
def lire_fichier_json(nom_fichier):  
    with open(nom_fichier, 'r', encoding='utf-8') as fichier_json:  
        return json.load(fichier_json)  
  
def generer_couleurs(nb):  
    return [QColor(random.randint(0, 255), random.randint(0, 255), random.randint(0, 255)) for _ in range(nb)]
```

Problématiques rencontrées

Mauvais tri des fichiers par taille : on a utilisé `split(",")` pour extraire la taille. Mais on a simplifié cela avec 2 variables. (i et m)

```
def listchem(doss):  
    liste = doss.rglob("*")  
    listtout = []  
    for i in liste:  
        if i.is_file():  
            i = i.resolve()  
            m = i.stat().st_size  
            listtout.append([str(i), m])  
    return listtout
```


Conclusion

Le Programme remplit bien les fonctions attendues (Analyser, trier et supprimer).

- Il arrive a trouver le chemin du fichier sélectionné
- Il arrive a trier la liste des sous fichiers dans l'ordre décroissant
- Il génère bien un script PowerShell de suppression

Intérêts de l'outil : Localiser et supprimer des fichiers volumineux de manière plus efficace.