

SUPPORT DE TPs

Les Fichiers & les exceptions

LES FICHIERS & LES EXCEPTIONS

Exercice 1 :

Dans cet exercice, vous allez travailler avec un fichier de logs réseau nommé `network_log.txt` contenant des informations sur les connexions réseau (adresses IP, ports, timestamps, etc.).

Il s'agit de développer un programme Python qui analyse ce fichier de logs pour extraire des informations importantes.

Étapes détaillées :

1. Développez un programme Python qui :
 - Compte le nombre total de lignes dans le fichier
 - Compte le nombre de connexions par protocole (HTTP, HTTPS, DNS, etc.)
 - Identifie toutes les adresses IP uniques et les stocke dans un ensemble (set)
 - Recherche toutes les connexions impliquant une adresse IP spécifique entrée par l'utilisateur
2. Utilisez la gestion d'exceptions pour :
 - Vérifier l'existence du fichier
 - Gérer les erreurs de lecture
 - Gérer les entrées incorrectes de l'utilisateur
3. Structure du programme (script fourni)

Exercice 2 :

Développer un scanner de ports simple qui vérifie quels ports sont ouverts sur une adresse IP donnée, puis sauvegarde les résultats dans un fichier.

Étapes détaillées :

1. Importation des modules nécessaires :

```
import socket
import os
import time
```

```
import json  
from datetime import datetime
```

2. Développez les fonctions principales :

- Fonction pour scanner un port spécifique
- Fonction pour scanner une plage de ports
- Fonction pour sauvegarder les résultats dans un fichier JSON
- Fonction pour charger et comparer des résultats précédents

Exercice 3 :

Développer un outil qui découvre les appareils sur le réseau local et crée une cartographie du réseau en sauvegardant les résultats dans différents formats (TXT, CSV, JSON).

Modules et structure :

1. Importation des modules nécessaires :

```
import subprocess  
  
import platform  
  
import os  
  
import csv  
  
import json  
  
import socket  
  
import ipaddress  
  
import concurrent.futures  
  
import time  
  
from datetime import datetime
```

2. Définition des structures de données

```
# Structure pour stocker les informations sur un appareil réseau

# Utilisation de tuples et de dictionnaires

def creer_appareil(ip, nom=None, mac=None, temps_reponse=None,
ports_ouverts=None):

    return {

        "ip": ip,

        "nom": nom,

        "mac": mac,

        "temps_reponse": temps_reponse,

        "ports_ouverts": ports_ouverts if ports_ouverts else [],

        "en_ligne": True if temps_reponse is not None else False

    }
```