

Mini projet Python fondamentaux (parcours cir)

Travail individuel

Objectif	
Synthèse des connaissances	
Ressources à disposition	
Le présent document liste les généralités pour l'évaluation	
Documents fournis :	
Un fichier csv	Contient les données à exploiter
<p>Au cours de votre travail, pensez à faire des commits ! Quand vous aurez terminé vous enverrez une copie de votre travail dans un fichier nommé '<i>nomPrenom.py</i>' par mail à olivier.gutierrez@limayrac.fr ATTENTION : une fois cela réalisé, vérifiez le contenu du fichier que vous avez envoyé.</p> <p>Vous devez tester de façon individuelle chaque fonction que vous créez avant d'utiliser cette fonction dans une autre fonction ou classe. Lors de l'écriture de code vous prendrez soin de donner des noms de variables représentatifs et de mettre des commentaires. Vous penserez aussi à libérer les ressources quand vous utiliserez des fichiers.</p>	

L'objectif final de votre travail est la création d'utilitaires qui vont simplifier votre travail d'administrateur système.

1. Mission

Votre boss vous demande de travailler sur des fichiers issus de postes utilisateurs, en l'occurrence il s'agit de fichiers au format .csv résultant de l'exécution d'une commande netstat -a

1.1 fichier à traiter

Voici un aperçu du type de fichiers que vous devez traiter (ici dans un tableur). Il possède 7 colonnes

A	B	C	D
Proto	Adresselocale	Adressedistante	Etat
TCP	0.0.0.0:135	AHMAD:0	LISTENING
TCP	0.0.0.0:445	AHMAD:0	LISTENING
TCP	0.0.0.0:3389	AHMAD:0	LISTENING
TCP	0.0.0.0:5040	AHMAD:0	LISTENING
TCP	0.0.0.0:30019	AHMAD:0	LISTENING
TCP	0.0.0.0:49664	AHMAD:0	LISTENING
TCP	0.0.0.0:49665	AHMAD:0	LISTENING
TCP	0.0.0.0:49666	AHMAD:0	LISTENING
TCP	0.0.0.0:49667	AHMAD:0	LISTENING
TCP	0.0.0.0:49668	AHMAD:0	LISTENING
TCP	0.0.0.0:49669	AHMAD:0	LISTENING
TCP	0.0.0.0:49670	AHMAD:0	LISTENING
TCP	0.0.0.0:49672	AHMAD:0	LISTENING
TCP	0.0.0.0:49699	AHMAD:0	LISTENING
TCP	127.0.0.1:8884	AHMAD:0	LISTENING
TCP	127.0.0.1:40712	AHMAD:0	LISTENING

Le même fichier au format CSV

```
Proto,Adresselocale,Adressedistante,État
TCP,0.0.0.0:135,AHMAD:0,LISTENING
TCP,0.0.0.0:445,AHMAD:0,LISTENING
TCP,0.0.0.0:3389,AHMAD:0,LISTENING
TCP,0.0.0.0:5040,AHMAD:0,LISTENING
TCP,0.0.0.0:30019,AHMAD:0,LISTENING
TCP,0.0.0.0:49664,AHMAD:0,LISTENING
TCP,0.0.0.0:49665,AHMAD:0,LISTENING
TCP,0.0.0.0:49666,AHMAD:0,LISTENING
TCP,0.0.0.0:49667,AHMAD:0,LISTENING
TCP,0.0.0.0:49668,AHMAD:0,LISTENING
TCP,0.0.0.0:49669,AHMAD:0,LISTENING
TCP,0.0.0.0:49670,AHMAD:0,LISTENING
TCP,0.0.0.0:49672,AHMAD:0,LISTENING
TCP,0.0.0.0:49699,AHMAD:0,LISTENING
TCP,127.0.0.1:8884,AHMAD:0,LISTENING
TCP,127.0.0.1:49712,AHMAD:0,LISTENING
TCP,127.0.0.1:49715,AHMAD:0,LISTENING
TCP,127.0.0.1:49716,AHMAD:0,LISTENING
TCP,127.0.0.1:49992,AHMAD:0,LISTENING
TCP,127.0.0.1:50029,AHMAD:50030,ESTABLISHED
TCP,127.0.0.1:50030,AHMAD:50029,ESTABLISHED
TCP,127.0.0.1:50031,AHMAD:50032,ESTABLISHED
TCP,127.0.0.1:50032,AHMAD:50031,ESTABLISHED
TCP,172.17.1.125:139,AHMAD:0,LISTENING
TCP,172.17.1.125:49713,zahn:30016,ESTABLISHED
TCP,172.17.1.125:49924,huxley:microsoft-ds,ESTABLISHED
TCP,172.17.1.125:49942,spinrad:epmap,ESTABLISHED
TCP,172.17.1.125:49943,spinrad:49692,ESTABLISHED
TCP,172.17.1.125:49947,spinrad:49692,ESTABLISHED
TCP,172.17.1.125:49949,spinrad:49692,ESTABLISHED
TCP,172.17.1.125:50984,zahn:30229,ESTABLISHED
```

2. Étapes du travail

2.1 Lire un fichier

Rédiger le code pour lire le fichier csv fourni :

1. afficher une '*' dans la console à chaque ligne lue
2. afficher le total de lignes du fichier
3. Il n'est pas demandé de gérer les erreurs qui pourraient se produire

Remarque : la première ligne du fichier à lire contient le nom des colonnes du fichier csv, vous aurez besoin de réutiliser ces noms par la suite

2.1.1 Création d'une liste globale

Créer une liste globale dans laquelle vous ajouterez toutes les listes créées à l'étape suivante. Cette liste globale sera affectée à une variable qui sera utilisée par la suite, elle sera nommée 'listeGlobale'

2.1.2 création d'une liste par ligne

Pendant la lecture du fichier, créer une list pour chaque ligne lue dans le fichier (utiliser la commande 'split')

2.2 Fonctions

2.2.1 Parcours de la 'listeGlobale'

Faire une fonction qui prend un argument.

1. Le premier est une liste du type de 'listeGlobale'
2. Elle utilise une boucle for pour afficher l'ensemble des listes présentes dans 'listeGlobale'

2.2.2 affichage des lignes correspondant à une valeur donnée

Nota : Les colonnes qui nous intéressent principalement sont Proto,Etat. On veut pouvoir afficher des informations selon le critère donné

Faire une fonction 'fonctionEgale' qui prend trois arguments :

1. Le premier est une liste du type de 'listeGlobale'
2. le 2eme est une valeur. Cette valeur correspond au numéro de la colonne que l'on veut traiter ('numCol').
3. le 3eme est une valeur qui va servir de filtre ('filtre')
4. La fonction utilise une boucle for pour parcourir les listes présentes dans 'listeGlobale'.
5. Quand la valeur de la colonne indiquée 'numCol' est égale à 'filtre' il faudra afficher la ligne correspondante

2.2.3 extraire une valeur de port d'une string

Nota : Soit la string '172.17.1.125:52224' ou 'a23-192-237-208:https'. On veut pouvoir récupérer la valeur avant ':' ou après

Faire une fonction 'prefixeSuffixe' qui prend deux arguments :

1. Le premier est une string contenant une chaine contenant des valeurs séparées par ':'
2. Le 2eme indique si la fonction doit retourner le préfixe ou le suffixe
3. la fonction retourne la valeur demandée

2.2.4 affichage des lignes correspondant à une valeur supérieure à un seuil

Nota : La colonne qui nous intéresse principalement est Adresselocale. On veut pouvoir afficher des informations selon le critère numérique donné. On veut notamment pouvoir afficher les lignes contenant un numéro de port supérieur à 49700

Faire une fonction 'fonctionSup' qui prend trois arguments :

1. Le premier est une liste du type de 'listeGlobale'
2. le 2eme est une valeur. Cette valeur correspond au numéro de la colonne que l'on veut traiter ('numCol').
3. le 3eme est une valeur qui va servir de limite ('seuil')
4. La fonction utilise une boucle for pour parcourir les listes présentes dans 'listeGlobale'.
5. Quand la valeur de la colonne indiquée 'numCol' est supérieure à 'seuil' il faudra afficher la ligne correspondante

2.2.5 affichage des lignes correspondant à une valeur supérieure, inférieure à un seuil ou comprise entre deux seuils

Nota : La colonne qui nous intéresse principalement est Adresselocale. On veut pouvoir afficher des informations selon le critère numérique donné en définissant la plage de valeurs

Faire une fonction 'fonctionPlage' qui prend quatre arguments:

1. selon les valeurs fournies, elle devra afficher toutes les valeurs :
 1. inférieures à un seuil
 2. supérieures à un seuil
 3. comprises entre deux seuils

2.2.6 Création d'un fichier

Nota : On veut pouvoir générer des fichiers différents en utilisant les fonctions créées au préalable. Ainsi on générera un fichier pour tous les protocoles 'TCP' ou un fichier pour toutes les ports inférieurs à 512, ou pour toutes les adresses distantes mentionnant 'http'

Faire une fonction qui prend plusieurs arguments :

1. Le premier est une liste du type de 'listeGlobale'
2. le 2eme est une String. Cette valeur correspond au nom de fichier qui sera créé
3. le 3eme permet de préciser à la fonction ce qu'elle devra faire (cad utiliser 'fonctionEgale' ou 'fonctionSup' ou 'fonctionPlage')
4. les arguments suivants sont ceux qui seront nécessaires pour alimenter les fonctions que l'on veut utiliser
5. La fonction utilise une boucle for pour parcourir les listes présentes dans 'listeGlobale'
6. Quand le travail d'écriture du fichier est terminé, il faut afficher son nom ainsi que le nombre de lignes qu'il contient

3. Scénarios de tests

Votre application , quand elle sera exécutée, devra montrer par des exemples que les fonctionnalités demandées sont fonctionnelles