Sciences Numériques et Technologie	2nde
Thème 4 : les données structurées et leur traitement Comment retrouver les identifiants d'une boîte mail par la programmation Python ?	TP

Contenus	Capacités attendues
Traitement de données structurées	Réaliser des opérations de recherche, filtre, tri ou calcul sur une ou plusieurs tables. Observer les différences de traitements possibles selon le logiciel choisi pour lire le fichier : tableur, programme Python

Dans ce TP nous allons réaliser un programme python pour retrouver les mots de passe et les login d'utilisateurs d'un service de messagerie à partir de données réparties sur deux fichiers.

ATTENTION : les fichiers *baselog.csv* et *basemail.csv* doivent être copiés dans *~/Mesdocuments/SNT/Donneés*

1. Traitement des données avec Python pour comprendre comment ça marche

1. Lire un fichier

Pour ouvrir un fichier, il faut commencer par créer un objet python f (par exemple) représentant le fichier créé et le mettant en état de lecture avec l'option 'r' et comme c'est l'option par défaut, il suffit d'écrire : f= open('nomFichier') où 'nomFichier' désigne le nom avec l'extension.
Pour le lire on utilise la méthode read ⇒ **f.read()**

Écrire l'exemple ci-dessous dans un script.
Enregistrer le programme dans : ~/Mesdocuments/SNT/Donneés ⇒ Nom du fichier : T4-TP-1.py
Puis l'exécuter.

```
f=open('baselog.csv',encoding='UTF8')
contenu=f.read()
f.close() # un fichier ouvert doit être toujours fermé à la fin du programme
```

Dans la console, taper les instructions suivantes :

```
>>> contenu          # permet d'afficher le contenu brut (/n veut dire retour à la ligne)
>>> type(contenu)    # permet de connaître le type de contenu (ici str = string = chaîne)
>>> print(contenu)   # permet d'afficher le contenu
```

On a ainsi récupérer le texte sous forme d'une chaîne de caractères, ce qui n'est pas très pratique pour traiter le fichier.

2. Lire un fichier CSV avec le module csv

Pour traiter des données numériques qui sont enregistrées dans un fichier .csv, il est fortement conseillé d'avoir enregistré le fichier au format csv avec l'encodage UTF-8 et un point pour le séparateur décimal si on souhaite faire des calculs avec les données.
Le module csv est un des modules qui permet de lire un fichier CSV et de le "transformer" en liste que l'on peut ensuite manipuler.

Écrire l'exemple ci-dessous dans un script.
Enregistrer le programme dans : ~/Mesdocuments/SNT/Donneés ⇒ Nom du fichier : T4-TP-2.py
Puis l'exécuter.

```
import csv
f=open('baselog.csv',encoding='UTF8')
r=csv.reader(f)
Log=list(r)
f.close()
```

Dans la console, taper les instructions suivantes :

```
>>>print(Log)   # permet d'afficher le contenu du csv
>>>type(Log)    # permet de connaître le type de contenu (ici list)
```

On récupère ainsi une liste de listes : la liste de toutes les lignes de la table, chaque ligne étant une liste et chaque élément de cette dernière est de type chaîne.
Plus précisément, un élément de `Log` obtenu par l'instruction `Log[i]` (où i est l'indice de l'élément) correspond à une ligne de la table.
L'instruction `len(Log)` qui renvoie la longueur de la liste Log correspond donc au nombre de lignes de la table. L'instruction `len(Log[0])` qui renvoie la longueur du premier élément de la liste `Log`, correspond donc au nombre de colonnes de la table. Enfin, une cellule de la table sera accessible par `Log[i][j]` où i est l'indice de la ligne et j celui de la colonne.

Dans la console, taper les instructions suivantes :

```
>>>nblignes=len(Log)
>>>nbcolonnes=len(Log[0])
>>>Ligne1=Log[0] ; print(Ligne1)   # définir et afficher les valeurs de la ligne 1
>>>Ligne3=Log[2] ; print(Ligne3)   # définir et afficher les valeurs de la ligne 3
```

Pour récupérer, par exemple, la première colonne autrement dit la liste des éléments d'indice 0 de chaque ligne qui dans notre exemple correspond à l'identifiant 'nom', on peut utiliser la fonction suivante (*à placer dans le code*) :

```
def Colonne(L,j):  #renvoie la liste des éléments de la colonne d'indice j de la table L
   C=[]   #on créé une liste C vide
   nblignes=len(L)    #nblignes correspond à la longueur de la table L
   for i in range(nblignes):    #répéter autant de fois que nblignes de la table L
      C.append(L[i][j])    #pour chaque ligne i on ajoute le contenu de la colonne j
   return(C)    #renvoie le résultat de la manipulation sur la liste C
```

Exécuter le code et dans la console taper l'instruction qui permet d'obtenir la liste des éléments de la première colonne de la table (Log).

```
>>>Colonne(Log,0)          # on affiche le contenu de la première colonne : les noms
>>>Colonne(Log,1)          # on affiche le contenu de la deuxième colonne : les logins
```

2. Croisement des données

1. À partir du fichier 'basemail.csv', obtenir la liste Mail correspondant à cette table.

Recopier les codes ci-dessous à la suite du script T4-TP-2.py - N'oubliez d'exécuter le script.

```
f = open('basemail.csv',encoding = 'UTF8')
r = csv.reader(f)
Mail = list(r)
f.close()
```

2. Vérifier que ces deux tables Log et Mail ont un descripteur en commun en affichant dans la console les deux listes contenant les descripteurs de celles-ci.

```
>>>print (Log) ; print (Mail)
```

3. On souhaite écrire une fonction loginPass(adressemail) qui prend en paramètre la chaîne adressemail et renvoie le login et le mot de passe correspondant.

La méthode index pour les listes permet d'obtenir l'indice d'un élément d'une liste.

```
def loginPass(adressemail) :
    CmailMail = Colonne(Mail,0) #on récupère la liste des éléments de la colonne
                                #qui contient les mails dans la liste Mail
    CnomMail = Colonne(Mail,1)      #on récupère la liste des éléments de la colonne
                                    #qui contient les noms dans la liste Mail
    indiceLmail = CmailMail.index(adressemail) #on obtient l'indice de la ligne
                                               #où se trouve adressemail dans CmailMail
    lenom = CnomMail[indiceLmail] #on obtient le nom correspondant à adressemail
    lemotdepasse = Mail[indiceLmail][2] #on obtient le mot de passe correspondant
                          #à adressemail dans le colonne 2 de la liste Mail
    CnomLog = Colonne(Log,0)    #on récupère la liste des éléments de la colonne
                                #qui contient les noms dans la liste Log
    indiceLlog = CnomLog.index(lenom)  #on obtient l'indice de la ligne
                                       #où se trouve lenom dans CnomLog
    lelogin = Log[indiceLlog][1] #on obtient le mot de passe correspondant au nom
    return(lelogin,lemotdepasse) #renvoie le résultat
```

N'oubliez d'exécuter le script.

4. Quel est le login et le mot de passe du détenteur de l'adresse 'mp@chez.moi' ?

Dans la console

>>>loginPass('mp@chez.moi')

Retrouver les logins et les mots de passe respectifs pour : jps@chez.moi et jrrt@chez.moi.