Programmation Python pour le Calcul Scientifique

Mohamed El Ghmary
mohamed.elghmary@um5s.net.ma

***

Partie A : Langage Python

Chapitre 09: Les Fichiers

1. Généralités

- Un fichier de données est un ensemble d'informations stockées sur une mémoire de masse (disque dur, disquette, CD-ROM,…).
- Il existe deux façons de coder les informations stockées dans un fichier de données : En binaire, où les informations sont stockées sous leurs forme binaire brute. En ASCII (textes), où les informations sont codées en ASCII .
- Un fichier peut être accessible en lecture seule, ou en lecture/écriture. Pour pouvoir être consulté et/ou modifié, il doit être ouvert d'abord, puis fermé ensuite (Python se charge souvent de fermer le fichier lui-même quand tout est terminé).

2. Schéma d'utilisation d'un fichier

- Ouverture ou création
- Séquences d'opération de lecture, écriture et recherche
- Fermeture du fichier

3. Ouverture et fermeture

Tout fichier doit être ouvert avant toute opération de création et d'utilisation (lecture/ écriture). Cette opération permet d'associer le nom physique du fichier à l'identificateur du fichier ainsi déclaré dans le programme.

SYNTAXE :

```
id_fichier = open(nom_physique_du_fichier , mode_accès)
```

- **nom_physique_du _fichier**: c'est le nom ou le chemin du fichier figurant sur le disque à ouvrir ou à créer.
- **mode_ accès**: c'est une chaîne de caractères qui précise le mode d'ouverture :

mode_ accès (pour les fichiers TEXTES) :

- '' r '' lecture seule
- '' w '' écriture seule (destruction de l'ancienne version si elle existe)
- '' w+ '' lecture/écriture (destruction ancienne version si elle existe)
- '' r+ '' lecture/écriture d'un fichier existant (mise à jour), pas de création d'une nouvelle version.
- '' a+ '' lecture/écriture d'un fichier existant (mise à jour), pas de création d'une nouvelle version, le pointeur est positionné à la fin du fichier.

mode_ accès (pour les fichiers binaires) :

- '' rb '' lecture seule
- '' wb '' écriture seule (destruction de l'ancienne version si elle existe)
- '' wb+ '' lecture/écriture (destruction ancienne version si elle existe)
- '' rb+ '' lecture/écriture d'un fichier existant (mise à jour), pas de création d'une nouvelle version.
- '' ab+ '' lecture/écriture d'un fichier existant (mise à jour), pas de création d'une nouvelle version, le pointeur est positionné à la fin du fichier.

Remarque :

- A l'ouverture, le **pointeur** est positionné **au début** du fichier (sauf '' a+ '' et '' ab+ '')
- Il faut toujours fermer un fichier à la fin d'un traitement.

SYNTAXE :

```
Id_fichier.close( )
```

4. Lecture d'un fichier texte :

- ***Id_fichier*.read(n)** (n caractères, et par défaut la totalité du fichier) : le résultat est une chaîne de caractères.
- ***Id_fichier*.readline()** renvoie une ligne de texte, terminée par le caractère \n de fin de ligne.
- ***Id_fichier*.readlines()** renvoie la liste de toutes les lignes de texte.

5. **Écriture dans un fichier texte :**
   - ***Id_fichier*.write(chaine)** écrit une chaîne de caractères dans un fichier texte.
6. **Lecture séquentielle d'un fichier texte :**

La syntaxe ***for ligne in Id_fichier*** permet de lire séquentiellement les différentes lignes d'un fichier texte. Cette syntaxe est supérieure à myfile.readlines(), qui charge la totalité du fichier en mémoire sous la forme d'une liste de chaînes.

7. **Exemple :**

```
equations = open("système.txt",”w”)
equations.write("X1+3X5-5X4=12\n")
equations.close()
```

8. **Lecture d'un fichier binaire :**
   - *info=load(Id_fichier*)
9. **Écriture dans un fichier binaire :**
   - *dump(info,Id_fichier*) écrit une information dans un fichier binaire.
10. **Lecture séquentielle d'un fichier binaire :**

```
from pickle import *
f=open(‘’ nomfichier ‘’, ‘’rb’ ‘)
try :
        while 1 :
                info=load(f)
                print(info)
except :
     f.close()
```

Résumé

Les fichiers	
Méthode	**Signification**
f=open(‘…’,’..’)	permet d'ouvrir ou de créer un fichier
f **.read(n)**	permet de lire n caractères
f **.read()**	permet de lire la totalité du fichier
f **.readline()**	permet de lire et renvoie une ligne de texte .
f **.readlines()**	permet de lire et renvoie la liste de toutes les lignes de texte.
f.**write(chaine)**	permet d'écrire une chaîne de caractères dans un fichier texte.
info=load(f)	permet de lire une information à partir d'un fichier binaire.
dump(info,f)	permet d'écrire une information dans un fichier binaire.
f.close( )	permet de fermer un fichier .
……	……

Série 09 : Les fichiers

Exercice 01 :

Copier un fichier texte dans un autre :

1. Caractère par caractère
2. Ligne par ligne
3. La totalité du fichier

Exercice 02 :

1. Lire un fichier texte, avec un contrôle d'erreur: L'utilisateur saisit le nom du fichier, la machine retourne le listing du fichier s'il existe et un message d'erreur s'il n'existe pas.
2. Calculer et afficher le nombre de caractères d'un fichier ASCII (un fichier texte)

Exercice 03 :

1. Créer un fichier texte de 5 chaînes de caractères (chacune par ligne).
2. Ajouter une fiche (c'est à dire une chaîne de caractères) au fichier précédent
3. Rechercher une fiche dans le fichier précédent.

Exercice 04 (CNC 2012) **Chiffrement d'un fichier texte par l'algorithme RC4**

Il s'agit de reprendre l'algorithme RC4 pour chiffrer les lignes d'un fichier texte donné.

Pour ce faire on suppose avoir définie la fonction d'entête suivante : **RC4Chaine(Claire,Clef)**

Cette fonction a pour paramètres :

- **Claire** : Une chaine de caractères contenant le texte clair qu'on désire chiffrer
- **Clef** : Une chaine de caractères contenant la clé de chiffrement

La fonction **RC4Chaine** retourne une chaine de caractères contenant le texte chiffré par l'algorithme **RC4**, de la chaine Claire en utilisant la clé de chiffrement **Clef**.

❖ **Question : Chiffrement d'un fichier**

En utilisant la fonction RC4Chaine décrite ci dessus,

✍ Écrire la fonction de prototype **: def RC4Fichier(fich, Clef, fichChiffre) :**Cette fonction permet de chiffrer en utilisant la chaine **Clef**(2ème paramètre) les lignes du fichier texte de nom physique **fich** (le fichier **fich** en premier paramètre est supposé existant). Les lignes chiffrées du fichier **fich** seront mises dans un nouveau fichier texte de nom physique **fichChiffre**(3ème paramètre) sera crée dans la fonction.

Exercice 05

Le but de cet exercice est de réaliser une application de gestion de notes des élèves d'un centre de CPGE .

- Un **élève** sera représenté dans ce programme par : **numIns, nom, classe, niveau.**
- Une **matière** sera représentée dans ce programme par : **codeMat, intitulé,coeff.**
- Un **examen** sera représenté dans ce programme par : **numIns, codeMat ,note .**

Questions : Ecrire les fonctions suivantes :

- **def ajouterEtudiant (numIns, nom, classe, niveau, fichEtudiants)** qui permet d'ajouter un nouvel élève .
- **def afficherEtudiant (numIns, fichEtudiants)** qui permet d'afficher un éléve donné par son numIns
- **def ajouterMatiere (codeMat, intitulé,coeff, fichMatieres)** qui permet d'ajouter une nouvel matière .
- **def afficherMatiere (codeMat, fichMatieres)** qui permet d'afficher une matière donnée par son **codeMat**.
- **def ajouterNote (numIns, codeMat ,note, fichNotes)** qui permet d'ajouter une nouvel note .
- **def afficherNote (numIns ,codeMat, fichNotes)** qui permet d'afficher la note 'une matière donnée par son **codeMat** un etudiant donné par son numIns .