
Licence 1 Informatique – 2022/2023 – S2

Algorithmique et programmation Python

TD 9 : Fichiers textes

Rappel des consignes :

- Créez un dossier pour la séance.
- Regroupez les fichiers Python « .py » dans ce dossier.
- 1 exercice = 1 fichier Python
- Le nom du fichier doit être en relation avec le numéro de l'exercice (« exercice_1.py », etc.)
- Rédigez en parallèle les algorithmes pour chaque exercice en pseudo-code dans un fichier word unique.
- A la fin de l'heure, vous regroupez les fichiers du dossier (word unique et .py pour chaque exercice) dans une archive (zip, 7z, rar, autre...)
- A rendre sur votre espace Moodle.
- N'oubliez pas de commenter abondamment votre code source Python.

Préambule : Attention, pour que les fichiers créés et lus soient situés dans le même dossier que vos programmes « .py », vous devez spécifier au début de chaque programme votre répertoire de travail. Utilisez l'instruction :

```
import os  
os.chdir('C:/... votre dossier...')
```

!!\ A défaut, en l'absence de cette spécification, le programme place et cherche de lui-même les fichiers dans votre dossier utilisateur « c:/utilisateurs/votre_nom ».

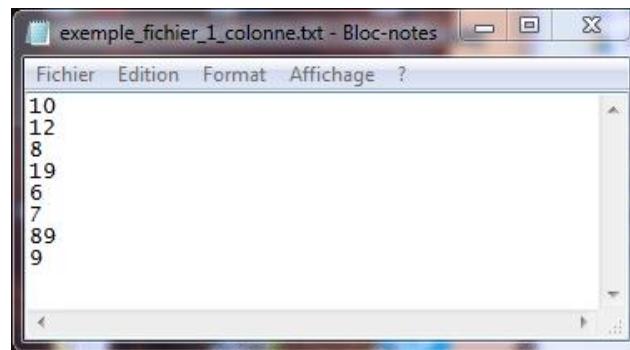
Exercice 1 – Ecriture d'un fichier de valeurs (fichier texte)

Créer une application qui :

- Demande à l'utilisateur le nombre d'éléments du tableau ;
- Procède à la saisie des **valeurs numériques entières** et les intègre dans une liste (créez un module, écrivez la fonctionnalité de saisie sous la forme d'une fonction qui prend en entrée le nombre d'éléments à saisir, elle renvoie les valeurs sous la forme d'une liste) ;
- Affiche les valeurs ;
- Demande à l'utilisateur un nom de fichier (extension « .txt ») ;

5. Sauvegarde le contenu du tableau dans le fichier texte. Pour cette opération, vous devez écrire une fonction qui prend en entrée la liste et le nom du fichier, laquelle doit être intégrée dans votre module.

Voici un exemple de fichier ouvert dans le bloc-notes de Windows



Remarque : la fonction `str()` permet de convertir un numérique en chaîne de caractères

Exercice 2 – Lecture d'un fichier de valeurs (fichier texte)

Créer une application qui :

1. Demande à l'utilisateur un nom de fichier texte censé contenir une liste de valeurs numériques entières ;
2. Charge les valeurs du fichier dans une liste de valeurs numériques, vous devez écrire une fonction spécifique pour cela [elle prend en entrée le nom du fichier, elle renvoie les valeurs dans une liste], que vous intégrerez dans le module de l'exercice précédent ;
3. Affiche les valeurs de la liste ;
4. Calcule et affiche la moyenne des valeurs.

Remarque (1) : la méthode `rstrip(caractère)` associée à la classe `String` permet de retirer un caractère spécifique situé à droite (en fin) d'une chaîne de caractère, s'il existe. Exemple :

```
s = « toto »  
z = s.rstrip(« o »)  
print(z) # renvoie la chaîne « tot »
```

Remarque (2) : Pour aller plus loin concernant les procédures de manipulation des chaînes de caractères, voir : <https://docs.python.org/3/library/stdtypes.html#string-methods>

Exercice 3 – Couples de valeurs, écriture

Créer une application qui :

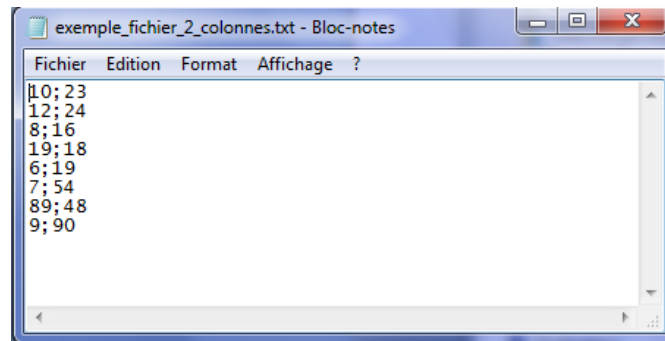
1. Demande à l'utilisateur le nombre d'éléments ;
2. Réalise la saisie de 2 listes de valeurs numériques X et Y ;
3. Calcule et affiche la somme du produit croisé entre ces deux vecteurs

$$S = \sum_i X_i * Y_i ;$$

Pour ce faire, créez une fonction qui prend en entrée deux listes et qui renvoie le résultat du calcul. Vous l'intégrez dans votre module.

4. Demande à l'utilisateur un nom de fichier ;
5. Sauvegarde les 2 vecteurs dans ce seul et même fichier (les colonnes seront séparées par le caractère « ; »). Ici aussi, créez une fonction à cet effet : elle prend en entrée les 2 listes et le nom du fichier de sauvegarde avec l'extension « .txt ».

Voici un exemple de fichier ouvert dans le bloc-notes de Windows.



Remarque : Il y a deux solutions possibles ici, sans ou avec le parseur « csv ». Tentez les deux approches.

Exercice 4 - Couples de valeurs, lecture

Créer une application qui :

1. Demande à l'utilisateur un nom de fichier contenant 2 colonnes de valeurs séparées par « ; » (cf. l'exemple ci-dessus) ;
2. Charge le contenu du fichier dans 2 listes d'entiers (écrivez la fonctionnalité sous la forme d'une fonction qui prend en entrée le nom de fichier, et qui renvoie les 2 listes, à vous de voir comment organiser au mieux la structure de sortie ; la fonction doit être intégrée dans votre module)
3. Affiche les deux listes
4. Affiche la somme des produits croisés entre les deux listes (cf. la formule ci-dessus ; si vous l'avez écrite sous forme de fonction, il suffit de faire l'appel en lui passant les deux listes issues du chargement de fichier).

Remarque : Peut-être que la fonction split() associée aux chaînes de caractères pourrait vous aider pour la décomposition de chaque ligne du fichier (cf. https://www.w3schools.com/python/ref_string_split.asp)

Remarque : Il y a deux solutions possibles ici, sans ou avec le parseur « csv ». Tentez les deux approches.