

TP (S1) 6

Lecture et écriture de fichiers

- 1 **Écriture de fichiers**
- 2 **Lecture d'un fichier**

Objectifs

- Savoir écrire un fichier.
- Savoir lire un fichier.

1. ÉCRITURE DE FICHIERS

Exercice 1 Création d'un fichier [Sol 1] On désire écrire une fonction `listeClasse(N:int) ->None` dont le but est de créer un fichier `liste.txt` contenant les prénoms de N élèves de la classe. On se limitera à un nombre d'élèves faible (N =4 par exemple).

1. Exécuter et commenter le code suivant. On remarquera, notamment, l'utilisation de `input(message:str) ->str` qui permet de stocker dans une variable la saisie au clavier de l'utilisateur après l'affichage d'un message

```
N = 4
liste = []
for i in range(N):
    liste.append(input('Quel est le prénom de l élève '+str(i+1)+' '))
print(liste)
```

2. Modifier le code précédent pour que le retour à la ligne `\n` soit accolé à chacun des prénoms.
3. Proposer une fonction `listeClasse(N:int) ->None` qui crée le fichier demandé et vérifie le résultat à l'aide d'un éditeur de texte.
4. En ouvrant le fichier avec l'attribut `'a'`, ajouter un nouveau nom dans le fichier à la suite des noms déjà enregistrés.

Exercice 2 Table de multiplication [Sol 2]

1. Créer une fonction `tableMulti(x:int) ->None` qui crée un fichier `tablex.txt` (où x est remplacé par la valeur rentrée en argument de la fonction, par exemple `table18.txt`) et dont chaque ligne est de la forme `x*1 = ...`, `x*2 = ...`, en remplaçant x par sa valeur et en indiquant le résultat jusqu'à `x*100`.
2. Exécuter la fonction pour `x = 5`, vérifier le fichier obtenu à l'aide d'un éditeur de texte, puis, avec Python, rouvrir le fichier en mode append `'a'` et ajouter une ligne supplémentaire pour `101*x`.

2. LECTURE D'UN FICHIER

Exercice 3 Lecture de fichiers [Sol 3]

1. Le fichier `entier.txt` contient une liste d'entiers. Déterminer la somme de ces entiers. *Réponse : 377958*
2. Le fichier `lettre.txt` contient une lettre de recommandation pour un étudiant. Afficher le contenu de la lettre puis recommencer en n'affichant que les lignes impaires. *Il est possible que vous rencontriez des problèmes d'affichage des accents, dans ce cas, il faut ouvrir le fichier avec l'instruction `open('lettre.txt', 'r', encoding='utf-8')`*

Exercice 4 Traitement d'un fichier de notes [Sol 4] Le fichier `notes.txt` contient les notes à 3 devoirs surveillés d'une classe. Chacune des lignes concerne un étudiant différent et est organisée de la façon suivante : prénom note au devoir 1 note au devoir 2 note au devoir 3. Par exemple : Thomas 12.5 17.3 8.

1. Pour éviter la correction fastidieuse d'un quatrième devoir, l'enseignant choisit d'attribuer à chaque étudiant une note supplémentaire tirée au hasard entre 0 et 20 (et arrondie au dixième de point). On pourra pour cela utiliser `round(np.random.random()*20,1)` après avoir importé le module `numpy`.

Écrire une fonction `ajoutNote(nomDeFichier:str) ->None` qui crée un nouveau fichier `notes2.txt`, construit sur la même base que le fichier `notes.txt` mais contenant le prénom et les notes aux quatre devoirs.

2. La fonction `tableNotes(nomDeFichier:str) ->list` du fichier `tablenotes.py` renvoie une liste dont chaque élément est une liste contenant : le prénom de l'étudiant (de type `str`) et ses notes à chaque devoir (de types `float`).

Pour information, cette fonction utilise la méthode `s.split(c)` qui permet de séparer une chaîne `s` autour d'un caractère particulier `c` (par défaut le caractère d'espacement `' '`):

```
>>> s = "abc 13 15 16"
>>> s.split()
['abc', '13', '15', '16']
```

Après avoir testé la fonction `tableNotes`, écrire une fonction `calculmoyennes(tableau:list) ->list` qui, à partir du tableau de notes généré par la fonction `tableNotes`, renvoie une liste contenant uniquement la moyenne (arrondie au dixième) de chaque étudiant (de la forme `[moyenne1, moyenne2, moyenne3...]`).

3. Écrire une fonction `stats(nomDeFichier:str) ->(float, float)` qui renvoie la moyenne générale et l'écart type de la classe et qui trace l'histogramme des moyennes des étudiants. Appliquez cette fonction au fichier `notes2.txt`.

Vous pourrez, pour cela, vous appuyer sur l'exemple suivant :

```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.normal(12, 3, 48) #génère 48 nombres aléatoires |
↳ autour de 12
plt.hist(x, 20)
plt.show()
```

Solution 1

1. Liste des prénoms avec input

```
N = 4
liste = []
for i in range(N):
    liste.append(input('Quel est le prénom de l élève
'+str(i+1)+' '))
print(liste)
```

2. Il suffit de concaténer \n avec le résultat du input

```
N = 4
liste = []
for i in range(N):
    liste.append(input('Quel est le prénom de l élève
'+str(i+1)+' ')+'\n')
print(liste)
```

3. Définition de la fonction

```
def listeClasse(N:int)->None:
    f = open('liste.txt','w')
    liste = []
    for i in range(N):
        liste.append(input('Quel est le prénom de l élève
'+str(i+1)+' ')+'\n')
    f.writelines(liste)
    f.close()
```

4. ouverture du fichier en mode append

```
f = open('liste.txt','a')
f.write('Thomas\n')
f.close()
```

Solution 2

- 1.
- `def tableMulti(x:int)->None:`

```
nomFichier = 'table'+str(x)+'.txt'
f = open(nomFichier,'w')
for i in range(1,101):
    ligne = str(x)+'x'+str(i)+'='+str(x*i)+'\n'
```

```
f.write(ligne)
f.close()
return()
```

- 2.
- `f = open('table12.txt','a')`
-
- `f.write('12*101='+str(101*12)+'\n')`
-
- `f.close`

Solution 3

1. Sommes des entiers du fichiers

```
f = open("entiers.txt",'r')
listeNombre = f.readlines()
f.close()
S = 0
for nbre in listeNombre:
    S += int(nbre[:-1])
print(S)
```

2. Lecture de la totalité de la lettre puis d'une ligne sur deux

```
f = open("lettre.txt",'r')
lettreComplete = f.read()
f.close()
print(lettreComplete)
```

```
f = open("lettre.txt",'r')
lignes = f.readlines()
f.close()
lettrepartielle = ''
for i in range(0,len(lignes),2):
    lettrepartielle += lignes[i][:-1]
print(lettrepartielle)
```

Solution 4

- 1.
- `def ajoutNote(nomDeFichier:str)->None:`

```
f = open(nomDeFichier,'r')
f2 = open("notes2.txt",'w')
for ligne in f:
    nouvelnote = round(np.random.random()*20,1)
    f2.write(ligne[:-1] + ' '+ str(nouvelnote) +'\n')
f2.close()
```

```
return()

2. def MoyenneElts(L:list)->float:
    #Fonction intermédiaire calculant la moyenne des éléments \
    ↪ d'une liste
    S = 0
    for element in L:
        S += element
    moyenne = S/len(L)
    return moyenne

def calcMoyenne(tableauNotes:list)->list:
    tmoyenne = []
    for elt in tableauNotes:
        ↪ tmoyenne.append(round(MoyenneElts(elt[1:]), 1))
    return(tmoyenne)

3. def ecartType(listenote:list)->float:
    moyenne = MoyenneElts(listenote)
    E = 0
    for elt in listenote:
        E += (elt-moyenne)**2
    return(round((E/len(listenote))**(1/2),1))

def stats(nomDeFichier:str)->list:
    tableau = tableaunotes(nomDeFichier)
    moyennes = calcMoyenne(tableau)
    print(moyennes)
    Moyenne = MoyenneElts(moyennes)
    Ect = ecartType(moyennes)
    plt.hist(moyennes,20)
    plt.show()
    return([Moyenne, Ect])
```