

Chapitre II

Lire et lecture des fichiers en Python

II.1 Introduction

Un fichier (*file*) est un ensemble de données stocké en général sur un support externe (disque dur, Clé USB, ...). Le fichier est accessible via son nom et/ou son répertoire, son contenu peut être lu ou modifié en utilisant des fonctions appropriées. Dans ce chapitre, nous exploiterons comment lire, écrire et manipuler les données dans un fichier.

II.2 Lecture dans un fichier

La lecture du contenu d'un fichier se fait en trois étapes :

- 1- L'ouverture du fichier en utilisant la fonction `open` suivi de nom du fichier, l'extension `.txt`, et la lettre `r` (cette lettre indique que le fichier est ouvert pour la lecture).
- 2- Appel à une fonction permet de lire le contenu du fichier (`read()`, `readlines()`, `readline ()...`).
- 3- Fermeture du fichier.

Exemple II.1

Supposant on a un fichier nommé 'chahid' (enregistré dans le bureau d'ordinateur) et contient les informations suivantes :

Rabah Bitat
Mohamed Boudiaf
Krim Belkacem
Larbi Ben M'Hidi
Mostefa Ben Boulaïd
Didouche Mourad

Le programme suivant lit le contenu du fichier 'chahid'

```
2 from os import chdir
3
4 # Importation de la fonction chdir (change change directory)
5 # a partir du module os (operating system )
6
7 chdir("C:/Users/Zorig Abdelmalik/Desktop")# Répertoire du fichier 'chahid'
8
9 Fil1 = open ("chahid.txt", "r") # Ouverture du fichier en mode lecture 'read'
10 # et affecter son contenu à un objet nommé Fil1
11
12
13 Fil1.readlines() # Lecture toutes les lignes du fichier 'chahid'
14
15 Fil1.close() # fermeture de l'objet 'Fil1'
16
```

Résultat d'exécution

```
Out[82]:
['Rabah Bitat\n',
'Mohamed Boudiaf\n',
'Krim Belkacem\n',
"Larbi Ben M'Hidi\n",
'Mostefa Ben Boulaïd\n',
'Didouche Mourad']
```

Dans la ligne 2 (*from os import...*), nous avons importé la fonction 'chdir' (change directory) a partir du module `os` (operating system). Cette dernière permet de changer le répertoire.

Dans la ligne 7, nous avons indiqué le répertoire du fichier 'chdir'.

Dans la ligne 9, nous avons ouvert le fichier 'chdir' en mode lecture (*r*) et affecter son contenu à un objet nommé 'Fil1'

Dans la ligne 13, nous avons appelé la fonction `readlines()` pour lire et afficher toutes les lignes du fichier 'chahid'.

Dans la ligne 15, nous avons appelé la fonction `close()` pour fermer l'objet 'Fil1'.

Remarque

Il existe d'autres méthodes peuvent être utilisées pour lire le contenu d'un fichier :

- La méthode `read()`, sans argument cette fonction lit la totalité du fichier et le renvoie sous forme une chaîne de caractères unique. De plus, elle peut également être utilisée avec argument (nombre entier) qui indiquera le nombre de caractères à lire, par exemple `read(20)`.
- La méthode `readline()` lit une ligne d'un fichier et la renvoie sous forme une chaîne de caractères
- La méthode `readlines()` lit toutes les lignes d'un fichier et les renvoie sous forme une liste de chaîne de caractères.

II.3 Mot-clé with

Le mot-clé with peut également être utilisée pour ouvrir les fichiers. De plus, à la fin du bloc d'indentation elle assure automatiquement la fermeture du fichier, dans ce cas l'appel à la fonction `close()` n'est pas nécessaire.

Exemple II.2

```
1
2 with open ("chahid.txt", "r") as fil1:
3     for i in fil1:
4         print(i)
5
```

Résultat d'exécution

```
Rabah Bitat
Mohamed Boudiaf
Krim Belkacem
Larbi Ben M'Hidi
Mostefa Ben Boulaïd
Didouche Mourad
```

Exemple II.3

Soient le fichier 'annee' qui contient les données suivantes :

```
1954
Révolution 54
1962
Indépendance
```

Le programme suivant lit et affiche le contenu et le type des données de chaque ligne

```
1
2 with open ("annee.txt", "r") as fil1:
3     for i in fil1:
4         print(i, type(i))
5
6
```

Résultat d'exécution

```
1954
<class 'str'>
Révolution 54
<class 'str'>
1962
<class 'str'>
Indépendance
<class 'str'>
<class 'str'>
```

On peut constater que python a considéré 1954 et 1962 comme des chaînes de caractères (*str*) et pas comme des entiers (*int*). Par conséquent, il est nécessaire de convertir les valeurs numériques s'il agit des opérations mathématiques.

II.4 Ecriture dans un fichier

L'écriture dans un fichier se fait généralement en trois étapes :

- 1- L'ouverture du fichier en utilisant la fonction *open* de manière similaire à celle utilisé précédemment pour lire un fichier sauf que on utilise la lettre *w* au lieu la lettre *r* (cette lettre indique que le fichier est ouvert pour l'écriture).
- 2- Appel à une fonction d'écriture dans les fichiers .txt (*write ()*).
- 3- Fermeture du fichier (sauf qu'on utilise l'instruction *with*)

Exemple II.4

Soient le fichier 'Spec_Elec' enregistré dans le répertoire 'C:\Users\zorig\Desktop'

Le programme suivant écrit dans le fichier Spec_Elec les données suivantes :

Instrumentation, Télécommunications, systèmes embarqués, Micro-electronique

```

8  from os import chdir
9  chdir("C:/Users/zorig/Desktop")
10
11 a=["Instrumentation", "STLC", "ESEM", "ME"]
12
13 with open("Spec_Elec.txt", 'w') as fil1:
14     for i in range( len(a) ):
15         fil1.write(a[i])

```

Ainsi on trouve dans le fichier Spec_Elec les données suivantes :



Pour écrire chaque élément de la liste 'a' dans une ligne séparée on utilise le séparateur de ligne (*\n*) comme suit :

```

8  from os import chdir
9  chdir("C:/Users/zorig/Desktop")
10
11 a=["Instrumentation", "STLC", "ESEM", "ME"]
12
13 with open("Spec_Elec.txt", 'w') as fil1:
14     for i in range( len(a) ):
15         fil1.write(a[i])
16         fil1.write("\n")

```

Ainsi on obtient le résultat suivant :

```

Spec_Elec - Bloc-notes
Fichier Edition Format Affichage ?
Instrumentation
STLC
ESEM
ME

```

Comme on peut utiliser l'écriture formatée comme suit :

```

8 from os import chdir
9 chdir("C:/Users/zorig/Desktop")
10
11 a=["Instrumentation", "STLC", "ESEM", "ME"]
12
13 with open("Spec_Elec.txt", 'w') as fill:
14     for i in range( len(a) ):
15         fill.write(f"** {a[i]} **\n")
16

```

Ainsi, on obtient le résultat suivant :

```

Spec_Elec - Bloc-notes
Fichier Edition Format Affichage ?
** Instrumentation **
** STLC **
** ESEM **
** ME **
|

```

II.4.1 Ecriture des valeurs numériques dans un fichier txt

II.4.1.1 utilisation de la fonction 'write()'

La fonction `write()` peut être utilisée pour écrire des valeurs numériques dans un fichier, cependant il est nécessaire de convertir leur types en `string` car la fonction `write()` ne peut recevoir que un string comme argument.

Exemple II.5

Le code ci-dessous montre comment utiliser la fonction `write()` pour écrire des valeurs numériques dans un fichier.

```

q=[3.14, 1e3, 0.707, 1e-2]
with open ("chahid.txt", "w") as fill:
    for i in range (len (q)):
        fill.write(str (q[i]) )
        fill.write("\n")

```

Remarque importante

Toutes les valeurs écrits par la fonction `write()` seront de type `string`.

II.4.1.1 utilisation des fonctions de module 'pickle'

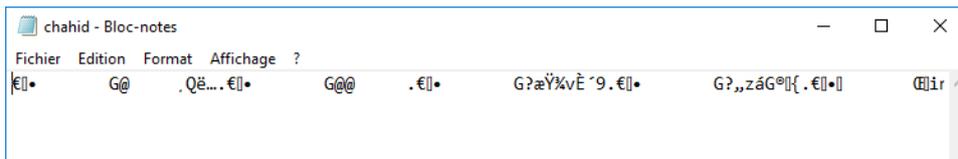
Une autre méthode pour écrire les valeurs numériques dans un fichier sans besoin de les convertir en string consiste à utiliser les fonctions du module `'pickle'`. Cependant les valeurs numériques seront enregistrées en format binaire.

Exemple II.6

```
import pickle

q= [3.14, 1e3, 0.707, 1e-2, "instrumentation", " Télécommunication" ]

with open ("chahid.txt", "wb") as fil1:
    for i in range (len (q)):
        pickle.dump( q[i], fil1 )
```

Résultat d'exécutionExemple II.7 (récupération des valeurs numériques)

```
with open ("chahid.txt", "rb") as fil2:
    for i in range(6):
        print (pickle.load(fil2))
```

Résultat d'exécution

```
3.14
1000.0
0.707
0.01
instrumentation
Telecommunication
```

II.5 Ouverture plusieurs fichiers en mode lecture et/ou écriture

Il est possible d'ouvrir plusieurs fichiers en même temps en utilisant le mot clé `with` de la façon suivante :

```
with open (" nom_du_fichier1. txt ", "r or w") as nom_objet1, open (" nom_du_fichier2.
txt ", "r or w") as nom_objet2,....., open (" nom_du_fichierN. txt ", "r or w") as
nom_objetN
```

Exemple II.8

Le programme suivant lit le contenu du fichier "File1" et l'écrit ensuite dans le fichier "File2"

```
2 from os import chdir
3 chdir("C:/Users/Zorig Abdelmalik/Desktop")
4
5 with open ("Fil1.txt", "r") as f1, open ("Fil2.txt", "w") as f2:
6     f2.write( f1.read() )
7
```

Exemple II.9

Le programme suivant lit ligne par ligne le contenu du fichier 'File1' et écrit les lignes qui ne commencent par le caractère '#' dans le fichier 'File2'.

```

1
2 from os import chdir
3 chdir("C:/Users/Zorig Abdelmalik/Desktop")
4
5 with open ("Fil1.txt", "r") as f1, open ("Fil2.txt", "w") as f2:
6     for i in f1:
7         if i[0] != "#":
8             f2.write(i)

```

II.6 Ajoute de lignes à un fichier existant

Il est possible d'ajouter des lignes à un fichier contient déjà des données en ouvrant le fichier en mode ajoute ('a'), dans ce cas les nouvelles lignes ajoutées seront placés après la dernière ligne.

Exemple II.10

Le programme suivant ajoute les éléments de la liste 'a' au contenu du fichier 'Fil1'

```

5 a=["Université de M'sila", "Département de L'electronique"]
6 with open ("Fil1.txt", "a") as f1:
7     for i in a:
8         f1.write(i)
9         f1.write("\n")

```

Contenu du fichier 'Fil1' avant l'exécution

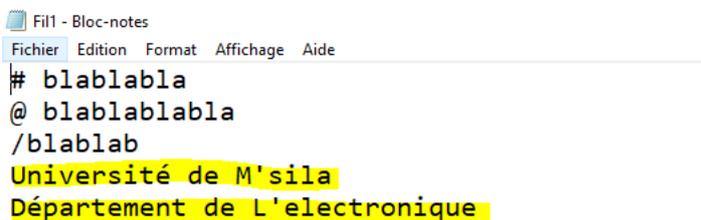


```

Fil1 - Bloc-notes
Fichier Edition Format Affichage Aide
# blablabla
@ blablablabla
/blablab

```

Contenu du fichier 'Fil1' après l'exécution



```

Fil1 - Bloc-notes
Fichier Edition Format Affichage Aide
# blablabla
@ blablablabla
/blablab
Université de M'sila
Département de L'electronique

```