Les fichiers

Introduction

- Lorsque on travaille avec les variables pour stocker les données, on remarque qu'après avoir terminé l'exécution du programme les valeurs prises par ces variables sont perdues. Car ces valeurs sont stockées dans la mémoire vive de l'ordinateur c'est-à-dire d'une façon temporaire.
- Pour stocker les valeurs de manière permanente c'est-à-dire sur disque dur, un des moyens c'est de travailler avec les **fichiers**.
- Le type de fichiers utilisés ici sont les fichiers texte (extension .txt).

Introduction

- Pour utiliser les fichiers dans un programme, il faudra :

1. **Ouvrir** le fichier dans un **mode** convenable c'est-à-dire pour écrire des données, ou lire des données
2. Faire **l'écriture** ou la **lecture** de ces données
3. **Fermer** le fichier

Ouvrir un fichier

- Pour ce faire on utilise la fonction **open()** de la manière suivante:
- Syntaxe:

```
fichier = open("nomfichier.txt", "mode")
```

- **mode**: c'est le mode d'ouverture du fichier
- **fichier**: c'est un objet de type file
- **Nomfichier.txt**: est le nom du fichier. Si le fichier se trouve dans un autre emplacement il faudra définir son chemin d'accès.

Les modes d'ouverture d'un fichier

1. **r**: pour une ouverture en **lecture** (**R**EAD).
2. **w** : pour une ouverture en **écriture** (**W**RITE), à chaque ouverture le contenu du fichier est **écrasé**. Si le fichier n'existe pas python le **crée**.
3. **a** : pour une ouverture en mode **ajout** à la fin du fichier (**A**PPEND). Si le fichier n'existe pas python le crée.

Exemples

- **Exemple 1:** le fichier est crée dans le même répertoire où se trouve mon programme.

```
fichier = open("MesDonnes.txt", "w")
```

- **Exemple 2:** le fichier est crée dans le chemin spécifié

```
fichier = open("D:/mondossier/MesDonnes.txt", "w")
```

- **Exemple 3:** on sort du répertoire en cours et on crée le fichier

```
fichier = open("../MesDonnes.txt", "w")
```

Ecrire dans un fichier

- Pour écrire dans un fichier, on utilise la méthode **write(chaine)**:

```
fichier = open("MesDonnees.txt", "a")
fichier.write("Bonjour monde")
```

- A noter que pour le mode d'ouverture **a** , si vous voulez écrire à la ligne, vous pouvez utiliser le saut de ligne **\n** :

```
fichier.write("Bonjour monde\n")
```

Fermeture d'un fichier

- Comme tout élément ouvert, il faut le fermer une fois les instructions terminées. Pour cela on utilise la méthode **close()** .

```
fichier.close( )
```

Lire le contenu d'un fichier

- Pour afficher **tout le contenu** d'un fichier ,vous pouvez utiliser la méthode **read()** sur le fichier. Cette méthode **retourne une chaine de caractère**.

```
fichier = open("MesDonnes.txt", "r")
print (fichier.read())
fichier.close()
```

- La méthode **read()** peut également être utilisée avec un argument. Celui-ci indiquera combien de caractères doivent être lus, à partir de la position déjà atteinte dans le fichier :

```
fichier = open("MesDonnes.txt", "r")
A=fichier.read(5)     # lit les 5 premiers caractères
Print(A)
B=fichier.read(15)    #lit les 15 caractères après les  5 premiers
Print(B)
fichier.close()
```

Lire toutes les lignes d'un fichier

- Il existe une autre méthode utile servant à lire les données du fichier c'est la méthode readlines().
- Cette méthode récupère **les lignes dans une liste**, chaque ligne est un élément de la liste résultante.

```
f=open("MesDonne.txt","r")
liste=f.readlines()

for   x  in liste:
    print(x)

f.close()
```

Lire une ligne d'un fichier

- Pour lire une seule ligne on peut utiliser la fonction **readline()** qui retourne **une chaine :**

```
fichier=open("MesDonne.txt","r")
ch=fichier.readline()
print(ch)
fichier.close()
```

Lire toutes les lignes d'un fichier

- Pour lire toutes les lignes à l'aide de la fonction **readline()**, on dois utiliser une boucle **while**.

```
fichier=open("MesDonne.txt","r")

ch=fichier.readline()

While (ch!=""):
    print(ch,end="")
    ch=fichier.readline()

fichier.close()
```

Remarque

- Toutes les données lues à partir d'un fichier sont récupérées **sous forme de chaine de caractères.**
- Si vous avez des valeurs numériques (entières ou réelles) elles ont également récupérées sous forme de chaine, donc si vous en avez besoin pour des calcules n'oubliez pas de faire la conversion en à l'aide des fonctions **float()** et **int()**

Supprimer un fichier

- Pour supprimer un fichier en python on utilisant la fonction **os.remove().**

```
import os
os.remove("mesDonnes.txt")
```

- On doit importer le module de OS pour supprimer un fichier en python.

Renommer un fichier

- La fonction **os.rename()** peut être utilisée pour renommer un fichier en Python.

```
import os
os.rename(file_oldname , file_newname)
```

- **file_oldname** : l'ancien nom de fichier.
- **file_newname** : le nouveau nom de fichier.
- Exemple:

```
os.rename("fichier1.txt", "fichier2.txt")
```

- On doit obligatoirement importer le **module os**

Vérifier l'existence d'un fichier

- Si on essai d'ouvrir un fichier qui n'existe pas, cela va générer une erreur, donc si on veut lire à partir d'un fichier, c'est intéressant de vérifier son existence avant de faire la lecture.
- Exemple :

```
import os
if os.path.exists(" fichier1.txt ") :
    liste=f.readlines()
    print(liste)
else:
    print("ce fichier n'exs=iste pas")
```