IngeSUP - Cours 06 - I/O fichiers txt

Sommaire

Objectifs

- Etre capable d'ouvrir et de fermer un fichier texte ;
- Savoir écrire dans un fichier texte ;
- Utiliser les différentes méthodes de lecture d'un fichier texte.

Prérequis

Avant de réaliser ce notebook de cours, vous devez avoir préalablement visualisé le mimo suivant :

- Module 8 : Les entrées/sorties – Fichiers (https://courses.ionisx.com/courses/ref/m189/x/courseware/f410c456d2ac45fd95af0b82d8068c24/258bbde4

Ouverture et fermeture d'un fichier texte

En Python, nous utiliserons la fonction `open` pour ouvrir un fichier. Celle-ci requière deux arguments, le chemin d'accès du fichier à lire ainsi que le mode d'ouverture. On utilisera une variable qui permettra de stocker l'information.

Entrée [ ]:

```
file_in = open("exemple.txt","w")
```

Un paramètre primordial de la fonction `open` est le mode d'ouverture. celui-ci est une chaîne de caractère qui spécifie la façon dont on souhaite ouvrir le fichier.

Le mode choisi conditionnera ce que vous pourrez faire avec le fichier.

Voici les modes d'ouverture couramment utilisés :

- 'w' : utilisé pour écrire dans un fichier ;
- 'r' : utilisé pour lire un fichier ;
- 'x' : utilisé pour créer et écrire dans un nouveau fichier ;
- 'a' : utilisé pour écrire à la suite d'un fichier existant ;
- 'r+' : utilisé pour lire et écrire dans un fichier ;

L'instruction `close` est quand à elle utilisée pour fermer le fichier ouvert.

Entrée [ ]:

```
file_in.close()
```

> 📝 **Note**
>
> Il est important de noter que, lors de la création d'un fichier texte, si aucun chemin d'accès absolu n'est spécifié, le fichier sera crée dans le même répertoire que le notebook qui exécute la commande.

Ecriture dans un fichier

Nous avons vu que le mode `"w"` permettait d'écrire au début d'un fichier texte préalablement ouvert tandis que le mode `"a"` permet lui d'écrire à la suite du fichier.

C'est l'instruction `write` qui commande l'écriture dans le fichier, écriture de la chaîne de caractère entrée en argument :

Entrée [ ]:

```
file_in = open("exemple_ecriture.txt","w")
file_in.write("Nous écrivons à la première ligne du fichier")
file_in.close()
```

Pour rappel, le mode `a` permet lui d'écrire à la suite d'un fichier. Par exemple:

Entrée [ ]:

```
file_in = open("exemple_ecriture.txt","a")
file_in.write("Nous écrivons à la suite du fichier")
file_in.close()
```

Il est souvent nécessaire d'aller à la ligne lorsque l'on écrit dans un fichier texte. Pour cela, nous utiliserons le caractère spécial `\n` .

Entrée [ ]:

```
file_in = open("exemple_ecriture.txt","a")
file_in.write("Nous écrivons à la suite du fichier\n")
file_in.write("Nous sommes allés à la ligne et nous réécrivons")
file_in.close()
```

Lecture d'un fichier

Il existe 3 méthodes permettant de lire un fichier txt en Python :

- méthode `.readlines()` ;
- méthode `.read()` ;
- méthode `.readline()` .

Méthode `.readlines()` :

La méthode `.readlines()` agit sur l'objet (ie le fichier) en déplaçant le curseur de lecture du début à la fin du fichier, puis elle renvoie une liste contenant toutes les lignes du fichier.

Par exemple :

Entrée [ ]:

```
# Ecriture du fichier txt
file = open("exemple_lecture.txt","x")
file.write("Margaux\nThéo\nSimon\nSophie")
file.close()

# Lecture du fichier txt
file = open("exemple_lecture.txt","r")
contenu = file.readlines()
file.close()
print(contenu)

# Suppression du fichier txt
import os
os.remove("exemple_lecture.txt")
```

Méthode `.read()` :

La méthode `.read()` lit tout le contenu d'un fichier et renvoie une chaîne de caractères unique.

Entrée [ ]:

```
# Ecriture du fichier txt
file = open("exemple_lecture.txt","x")
file.write("Margaux\nThéo\nSimon\nSophie")
file.close()

# Lecture du fichier txt
file = open("exemple_lecture.txt","r")
contenu = file.read()
file.close()
print(contenu)

# Suppression du fichier txt
import os
os.remove("exemple_lecture.txt")
```

Méthode `.readline()` :

La méthode `.readline()` (sans s à la fin) lit une ligne d'un fichier et la renvoie sous forme de chaîne de caractères.

À chaque nouvel appel de `.readline()`, la ligne suivante est renvoyée.

Entrée [ ]:

```
# Ecriture du fichier txt
file = open("exemple_lecture.txt","x")
file.write("Margaux\nThéo\nSimon\nSophie")
file.close()

# Lecture du fichier txt
file = open("exemple_lecture.txt","r")
contenu = file.readline()
file.close()
print(contenu)

# Suppression du fichier txt
import os
os.remove("exemple_lecture.txt")
```

Ouverture d'un fichier avec la méthode `with`

Il existe en Python le mot-clé `with` qui permet d'ouvrir et de fermer un fichier de manière efficace.

Si pour une raison ou une autre l'ouverture ou la lecture du fichier conduit à une erreur, l'utilisation de with garantit la bonne fermeture du fichier, ce qui n'est pas le cas dans le code précédent. Voici donc le même exemple avec `with` :

Entrée [ ]:

```
# Ecriture du fichier txt
file = open("exemple_lecture.txt","x")
file.write("Margaux\nThéo\nSimon\nSophie")
file.close()

# Lecture du fichier txt
with open("exemple_lecture.txt", 'r') as filin:
    lignes = filin.readlines()
    for ligne in lignes:
        print(ligne)

# Suppression du fichier txt
import os
os.remove("exemple_lecture.txt")
```

Exercices de TD

Vous pouvez maintenant vous exercer à partir du notebook [TD 06 - I/O fichiers txt (../TD/TD%2005%20-%20IO%20fichiers%20txt.ipynb)](../TD/TD%2005%20-%20IO%20fichiers%20txt.ipynb).