

VI. Fichiers de données

Une manière de stocker des données de manière pérenne est de les stocker dans des fichiers.

1. Editer un fichier

Pour éditer un fichier en python on utilise la fonction **open**.

Cette fonction prend en premier paramètre le chemin du fichier (relatif ou absolu) et en second paramètre le type d'ouverture

2. Chemin relatif / chemin absolu

Un **chemin relatif** en informatique est un chemin qui prend en compte l'emplacement de lecture.

Un **chemin absolu** est un chemin complet qui peut être lu quelque soit l'emplacement de lecture.

3. La fonction open

Voici la syntaxe pour lire un fichier

```
>>> fichier = open("data.txt", "r")
>>> print(fichier)
<open file 'data.txt', mode 'r' at 0x7ff6cf3fe4b0>
```

On remarque que le deuxième paramètre est renseigné par un **r**, ce paramètre indique une ouverture de fichier en lecture.

⇒ *Les types d'ouverture*

Il existe plusieurs modes d'ouverture:

- **r**, pour une ouverture en lecture (READ).
- **w**, pour une ouverture en écriture (WRITE), à chaque ouverture le contenu du fichier est écrasé. Si le fichier n'existe pas python le crée.
- **a**, pour une ouverture en mode ajout à la fin du fichier (APPEND). Si le fichier n'existe pas python le crée.
- **b**, pour une ouverture en mode binaire.
- **t**, pour une ouverture en mode texte.
- **x**, crée un nouveau fichier et l'ouvre pour écriture

4. Fermeture d'un fichier

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

```
>>> fichier.close()
```

5. Lire le contenu d'un fichier

Pour afficher tout le contenu d'un fichier, vous pouvez utiliser la méthode **read** sur l'objet-fichier.

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

⇒ **readlines**

Le contenu du fichier est stocké dans une liste, une ligne = un élément. Le caractère `\n` est maintenu. Il n'est pas présent sur la dernière ligne de notre fichier exemple.

```
#ouverture en lecture
f = open("voitures.txt", "r") #lecture
lst = f.readlines()
#affichage
print("** contenu de lst **")
print(lst)
print("** fin contenu **")
f.close()
```

6. Ecrire dans un fichier

Voici la syntaxe pour écrire dans un fichier:

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

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 = open("data.txt", "a")
fichier.write("\nBonjour monde")
fichier.close()
```

⇒ **writelines**

Permet d'écrire directement le contenu d'une liste. Nous devons quand même insérer le caractère « `\n` » pour que le saut de ligne soit effectif dans le fichier.

```
#ouverture en écriture
f = open("moto.txt", "w")
#liste
L = ["PCSI 1\n", "PCSI 2\n", "PSI 1\n", "PSI 2\n"]
#écriture
f.writelines(L)
#fermeture
f.close()
```