

Python

Cours 7 - Lecture et Écriture de Fichiers Divers

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Les fichiers et le principe de stockage
- 2 Lecture de Fichiers (“Read”)
- 3 Écriture de Fichiers (“Write”)
- 4 Options de la fonction interne “open”

Les fichiers et le principe de stockage

Dans le domaine des sciences des données et de l'Intelligence Artificielle (AI), le moyen de transfert le plus facilement manipulable et efficace à transféré est avec l'aide des fichiers textes ou "Comma Seperated Values".

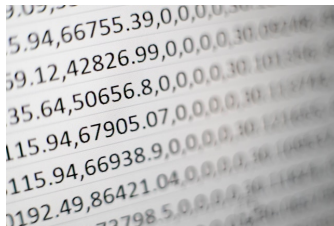


Figure: Information brute dans un fichier .txt ou .csv.

Les fichiers et le principe de stockage

Extensions de fichiers :

- Texte = .txt
- “Comma Seperated Values” = .csv

Lecture de Fichiers (“Read”)

La lecture de fichiers permet d'extraire l'information du fichier respectif et de le traiter grâce au langage de programmation Python.

Lorsque le fichier texte est en mémoire, toutes les opérations sur les chaînes de caractères, conversions, casting, etc., peuvent être appliquées.

Lecture de Fichiers (“Read”)

Attention : L'espace de mémoire vive de votre ordinateur (RAM) dicte la limite de mémoire que vous pouvez injecter dans une (ou plusieurs) variable lors de l'exécution de votre programme.

Si votre fichier texte fait 100 GB, naturellement, vous ne pouvez pas le charger en mémoire vive dans votre programme en exécution.

Lecture de Fichiers (“Read”)

```
1  # Steve Levesque, All rights reserved
2
3  with open('courses.txt', 'r') as reader:
4      for line in reader:
5          print(line, end='')
6
7  """
8  Python
9  NoSQL
10 Artificial Intelligence
11 """
```

Écriture de Fichiers (“Write”)

L'écriture de fichiers permet de mettre sur disque dur (HDD / SSD) les données créées par un programme lors de son exécution.

De nos jours, la mémoire disque est très ample et nous pouvons sauvegarder des Téraoctets de mémoire.

Écriture de Fichiers (“Write”)

```
1  # Steve Levesque, All rights reserved
2
3  with open('courses_newlist.txt', 'w') as writer:
4      writer.write("ReactJS")
5
6  # Inside courses_newlist.txt file
7  """
8  ReactJS
9  """
```

Écriture de Fichiers (“Write”)

Attention : Si votre programme injecte une grande quantité de données seulement à la toute fin de votre opération algorithmique, il se peut que votre ordinateur n'ait pas assez de mémoire vive pour le faire.

Conseil : Opter pour une approche **séquentielle** est mieux pour votre ordinateur dans des cas de figure larges. Dans le domaine de l'Intelligence Artificielle et du “Big Data” (Données massives), les quantités de données touchent toujours les centaines de GB et atteignent les Téraoctets.

Écriture de Fichiers (“Write”)

```
1  # Steve Levesque, All rights reserved
2
3  with open('courses_newlist.txt', 'a') as writer:
4      writer.write("\n" + "ReactJS")
5
6  # Inside courses_newlist.txt file
7  """
8  ReactJS
9  ReactJS
10 """
```

Options de la fonction interne “open”

Voici les options disponibles pour les opérations diverses sur les fichiers :

- 'r' - open for reading (default)
- 'w' - open for writing, truncating the file first
- 'x' - open for exclusive creation, failing if the file already exists
- 'a' - open for writing/append at end of the file if it exists
- 'b' - binary mode
- 't' - text mode (default)
- '+' - open a disk file for updating (reading and writing)

Options de la fonction interne “open” - Table

	r	r+	w	w+	a	a+
read	+	+		+		+
write		+	+	+	+	+
write after seek		+	+	+		
create			+	+	+	+
truncate			+	+		
position at start	+	+	+	+		
position at end					+	+

Figure: <https://stackoverflow.com/questions/1466000/difference-between-modes-a-a-w-w-and-r-in-built-in-open-function>

Options de la fonction interne “open” - Graphe

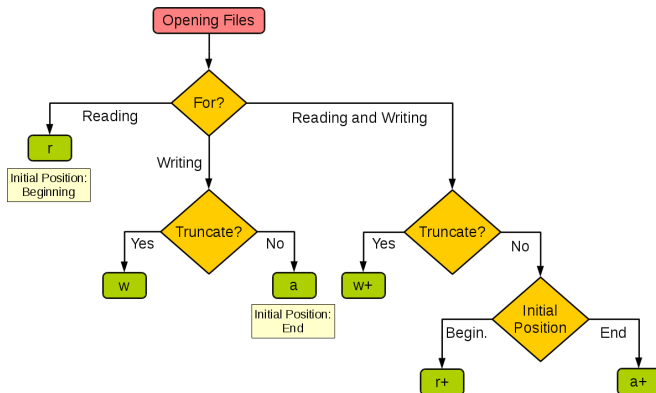


Figure: <https://stackoverflow.com/questions/1466000/difference-between-modes-a-a-w-w-and-r-in-built-in-open-function>

- <https://realpython.com/read-write-files-python/>