

Les fichiers avec Python

March 21, 2022

1 Introduction

Vous travaillerez sur des fichiers avec Python dans la SAE24. Ce notebook vous présente donc une introduction sur les formes de fichiers texte ou binaire ainsi que l'ouverture, la lecture et l'écriture dans un fichier avec Python.

2 Format de fichiers texte ou binaire

De manière générale, tous les fichiers peuvent être classés en deux grands formats : texte et binaire.

Les **fichiers binaires** peuvent stocker n'importe quel type de données comme l'image, la vidéo et l'audio dans le même fichier. C'est par exemple le cas des fichiers **png**, **pdf**, **ppt**, **mpeg**, La seule condition qu'ils présentent est d'avoir un programme approprié pour lire ce type de données présentes dans le système.

Les **fichiers texte** peuvent être considérés comme une version réduite des fichiers binaires, ne pouvant stocker que des données textuelles. C'est par exemple le cas des fichiers **txt**, **rtf**, **HTML**, **xml**, **json** ou **csv**. Tous les fichiers texte suivent la norme ASCII au minimum (ou un norme plus vaste englobant la norme ASCII comme UTF8), pour stocker des données. Les fichiers texte peuvent être visualisés par n'importe quel éditeur de texte.

3 Ouverture, lecture et fermeture d'un fichier

Les étapes d'ouverture et de fermeture sont toujours présentes en ce qui concerne les fichiers. Il s'agit d'indiquer au système d'exploitation que le programme souhaite accéder à un fichier et interdire à tout autre programme l'accès à ce fichier. Un autre programme qui souhaiterait créer un fichier du même nom ne le pourrait pas tant que l'étape de fermeture n'est pas exécutée. En revanche, il pourrait tout à fait le lire car la lecture ne perturbe pas l'écriture. Pour lire un fichier, il faut donc tout d'abord l'ouvrir avec la fonction **open(fichier, mode)** :

- **fichier** : chemin du fichier (relatif ou absolu)
- **mode** : le mode est composé de deux lettres, les droits d'accès (rwx) et l'encodage (b ou t). Par défaut le fichier est ouvert avec le mode 'rt'.
 - **r**, (par défaut)ouverture en lecture seule (READ).
 - **w**, ouverture en écriture (WRITE), à chaque ouverture le contenu du fichier est écrasé. Si le fichier n'existe pas, Python le crée.

- **a**, ouverture en écriture avec ajout (APPEND) à la fin du fichier, si il existe. Si le fichier n'existe pas, Python le crée.
- **x**, ouverture en écriture seulement si le fichier n'existe pas.
- **b**, pour une ouverture en mode binaire: les octets seront traités tels quels.
- **t**, (par défaut) pour une ouverture en mode texte: les octets seront traités comme des caractères affichables.

Une fois le fichier en lecture ouvert, on peut utiliser les méthodes:

- **read()** qui retourne une chaîne d'octets contenant l'intégralité du fichier,
- **read(x)** qui retourne une chaîne contenant les x prochains octets du fichier,
- **readline()** qui retourne une chaîne contenant la prochaine chaîne du fichier,
- **readlines()** qui retourne une liste de chaque ligne du fichier. Chaque ligne est terminée par "\n" qui représente un retour à la ligne. Cette fonction ne concerne que les fichiers texte.

Pour fermer le flux (le fichier) on utilise la méthode **close()** sur la variable qui le représente.

3.1 premier exemple avec un fichier texte:

```
[9]: print('Lecture du contenu du fichier Texte.txt qui est un fichier au format_
      ↳texte :\n')

fichier = open("Texte.txt", 'rt')
texte = fichier.read()
print('READ:')
print(texte)
fichier.close()

fichier = open("Texte.txt", 'rt')
texte = fichier.read(15)
print('READ(15):')
print(texte)
fichier.close()

fichier = open("Texte.txt", 'rt')
lignes = fichier.readline()
print('\nREADLINE:')
print(lignes)
fichier.close()

fichier = open("Texte.txt", 'rt')
lignes = fichier.readlines()
print('READLINES:')
print(lignes)
fichier.close()
```

Lecture du contenu du fichier Texte.txt qui est un fichier au format texte :

READ:

Ceci est un exemple de fichier texte.

Son objectif est de démontrer les différentes possibilités offertes par Python pour accéder à son contenu.

Un fichier texte ne contient par définition QUE des caractères affichables.

Les caractères affichables incluent également les caractères de tabulation, de fin de ligne, ou autres caractères spéciaux considérés néanmoins comme affichables.

READ(15):

Ceci est un exe

READLINE:

Ceci est un exemple de fichier texte.

READLINES:

```
['Ceci est un exemple de fichier texte.\n', 'Son objectif est de démontrer les  
différentes possibilités offertes par Python pour accéder à son contenu.\n', 'Un  
fichier texte ne contient par définition QUE des caractères affichables.\n',  
'\n', 'Les caractères affichables incluent également les caractères de  
tabulation, de fin de ligne, ou autres caractères spéciaux considérés néanmoins  
comme affichables.\n']
```

3.2 Deuxième exemple avec un fichier binaire (pdf):

```
[2]: print('Lecture du contenu du fichier Texte.pdf (pdf au format binaire) et  
      ↳affichage des 30 premières lignes :\n')  
fichier = open("Texte.pdf", 'rb')  
texte = fichier.readlines()  
print(texte[0:30])
```

Lecture du contenu du fichier Texte.pdf (pdf au format binaire) et affichage des 30 premières lignes :

```
[b'%PDF-1.3\n', b'%\xc4\xe5\xf2\xe5\xeb\xa7\xf3\xa0\xd0\xc4\xc6\n', b'4 0  
obj\n', b'<< /Length 5 0 R /Filter /FlateDecode >>\n', b'stream\n', b'x\x01\xadU  
\xcbn\xdb0\x10\xbc\xfb+\xa6\xb7\x06H\x15\x91zX\xba6\xc8\xad\x87\x1aU>\x80\xa6\xa  
\x84\x81D)" \x05\xb8\x7f\xd0\x93\xfa\xcb] \xc9F, \xc9H\xaa\x16\x86\x0f\xa4\x17\xc4  
\xee\xce\xec\xcc\xea\x15; \xbc\xe2\xee\xde2H\x0b\x7f\xfcYI! \xdf\xe3\xe1\xf1\xffpa  
\x1cQ\x92z} \x87\xac\x05\x03\x17\xdf\x03} \xce\x90I \x84\xf1\xf8\x8c\x0e\x1eF\x14  
\xf5\xe9\xa9\x1f\x0b2\n', b'wY\xc6o\n', b'| \xbewRCY\x87\xce@\x1dT\xd5\x94\n',  
b' \xb9B\xa1\xe5\xb3V-\x9c:8\xe5\xdd { \xc1C6v\xf4N\xf9\xcd\xb1\xfc\xb9& \xe3q\xe4\  
\xa5\xdbd\xa8\x89yM\xac\xcf\xf71\x9c0" \x1038?j \x83z\xff\xa2\xa4\xd3\xc5\x08\x8b\x  
b0\xe4}U \x1b\xd7\x12\x9aRY\xe4\xba( \xfaV\x19G\xf7\xa6\xb6V\xefu\xa9] oQ\x17\x85j\  
\xc7\xa8hq\xb3\xf9+\xe2U\x84\x07i \xb2
```

```

\xfc\xfb0\xf7LM6u\xd7BH\xd9\xe7\xd4\xd7/X\n',
b'I\xeaR\x99\xee\xff\xd8fI\xe0\xb14\x06\x15\\Pr5\xb6\x83p9\xc8G3\x17\n',
b'\x8c\x1aQh\xa2\x17\r\xd1\x98\xf7\x856\xdaiB\xb7{| aYH\xd1\n', b"\xe9~\xb7t\xbd
\x16\xc9<e\x0b\x92E1\x08X\xeci\xe0k\xe8<\x8d\xf2,\xdeh\x1b{\x8c'\x1c\x97\xa9\xd7
\xd0y\x91\xef\xcd\x80<X\xea\xei\x1f\xf2}l\x06\x96,\xbd\xfdmN\xf7\x84\x14h#\xcbn\
x98R\xff$JU\r\xb7\xc1\x1c\x93\xe1\x90q\xae5\x1f\x16,\xe7\xe3\xc4\xbe+\xc5\xa0\x8
b\xdb\xe3\xb61\xc3Q\xea'\xa3nQw\x10\x1d\xf9u\xd6\x8emz\xa9Ew\x18\xf4eu\xde\xb7\x
e4\xd8k\xf5GK*\xab6\x93\xa5hza\xaaZ\x1b\xea\xa0\xae*\x85\tsk\xe4t\xb9\x0b}\xb6\x
f5\xe2\x80c\xac4]\xbfk\xa6\x7fJ\x17\x1d\xbf\x01t\xbc\xa9\x89\xa5'\xaf\x07\xa7m\x
fe\xe9\xbcYw\x7f\x00\x14W\x8ah\n", b'endstream\n', b'endobj\n', b'5 0 obj\n',
b'490\n', b'endobj\n', b'2 0 obj\n', b'<< /Type /Page /Parent 3 0 R /Resources 6
0 R /Contents 4 0 R /MediaBox [0 0 612 792]\n', b'>>\n', b'endobj\n', b'6 0
obj\n', b'<< /ProcSet [ /PDF /Text ] /ColorSpace << /Cs1 7 0 R >> /Font << /TT3
10 0 R\n', b'/TT1 8 0 R >> >>\n', b'endobj\n', b'11 0 obj\n', b'<< /Length 12 0
R /N 3 /Alternate /DeviceRGB /Filter /FlateDecode >>\n', b'stream\n',
b'x\x01\x9d\x96wTS\xd9\x16\x87\xcf\xbd7\xbd\xd0\x12" %\xf4\x1az\t \xd2;H\x15\x04
Q\x89I\x80P\x02\x86\x84&vD\x05F\x14\x11)VdT\xc0\x01G\x87"cE\x14\x0b\x83\x82b\xd7
\t\xf2\x10P\xc6\xc1QDE\xe5\xdd\x8ck\t\xef\xad5\xf3\xde\x9a\xfd\xc7Y\xdf\xd9\xe7\
xb7\xd7\xd9g\xef}\xd7\xba\x00P\xfc\x82\x04\xc2tX\x01\x804\xa1X\x14\xee\xeb\xc1\
\x12\x13\xcb\xc4\xf7\x02\x18\x10\x01\x0eX\x01\xc0\xe1ff\x04G\xf8D\x02\xd4\xfc\x
b d=\x99\x99\xa8H\xc6\xb3\xf6\xee.\x80d\xbb\xdb,\xbfp&s\xd6\xff\x7f\x91"7C$\x06\x0
0\n', b'E\xdd56<~&\x17\xe5\x02\x94S\xb3\xc5\x192\xff\x04\xca\xf4\x95)2\x8612\x16\
xa1\t\xa2\xac"\xe3\xc4\xaf1\xf6\xa7\xe6+\xbb\xc9\x98\x97&\xe4\xa1\x1aY\xce\x19\x
bc4\x9e\x8c\xbbP\xde\x9a%\xe1\xa3\x8c\x04\xa1\\\x98%\xe0g\xa3|\x07e\xbdTI\x9a\x0
0\xe5\xf7(\xd3\xd3\xf8\x9cL\x000\x14\x99_\xcc\xe7&\xa11\x892E\x14\x19\xee\x89\xf
2\x02\x00\x08\x94\xc49\xbcR\x0e\x8b\xf99h\x9e\x00x\xa6g\xe4\x8a\x04\x89Ib\xa6\x1
1\xd7\x98i\xe5\xe8\xc8f\xfa\xf1\xb3S\xf9b1+\x94\xc3M\xe1\x88xL\xcf\xf4\xb4\x0c\x
8e0\x17\x80\xafo\x96E\x01%Ym\x99h\x91\xed\xad\x1c\xed\xedY\xd6\xe6h\xf9\xbf\xd9\
xdf\x1e~S\xfd=\xc8z\xfbU\xf1&\xec\xcf\x9eA\x8c\x9eY\xdf1\xec\xac/\xbd\x16\x00\xf
6$Z\x9b\x1d\xb3\xbe\x95U\x00\xb4m\x06@\xe5\xe1\xac0\xef \x00\xf2\x05\x00\xb4\xde
\x9c\xf3\x1e\x861~\x92\xc4\xe2\x0c'\x0b\x8b\xec\xec1s\x01\x9fk.+ \xe87\xfb\x9f\x
82o\xca\xbf\x869\xf7\x99\xcb\xee\xfbV;\xa6\x17?\x81#I\x153eE\xe5\xa6\xa7\xa6KD\x
cc\xcc\x0c\x0e\x97\xcfdf\xfd\xf7\x10\xff\xe3\xc09i\xcd\xc9\xc3,\x9c\x9f\xc0\x17\x
f1\x85\xe8UQ\xe8\x94\t\x84\x89h\xbb\x85<\x81X\x90.d\n']

```

Vous remarquerez que chaque chaîne commence par la lettre ‘b’ (qui signifie affichage binaire), ce qui prévient que des octets non affichables peuvent être présents. Dans ce cas ils seront affichés en hexadécimal.

Une valeur hexadécimale est précédée par les deux symboles: \x

Tout octet affichable sera représenté par le caractère correspondant.

Par exemple, vous ne verrez jamais \x41 puisque cette valeur hexadécimale (65 en décimal) est le code ASCII de la lettre ‘A’. La lettre ‘A’ sera donc affichée à la place.

4 Ecriture dans un fichier

On peut écrire dans un fichier si celui-ci est ouvert en écriture. Les trois modes possibles sont “w”, “x” et “a”.

À l'instar de `read()` et `readlines()`, on utilisera: - **`write(chaine)`** pour écrire la chaîne de caractères dans le fichier - **`writelines(lignes)`** pour écrire la liste de lignes dans le fichier.

A titre d'exemple, le programme ci-dessous a été exécuté 3 fois de suite:

```
[8]: fichier = open("Texte2.txt", 'a')
fichier.write(" \\nAvec le mode 'a' le contenu du fichier n'est pas effacé ...")
fichier.close()
fichier = open("Texte2.txt", 'rt')
texte = fichier.read()
print(texte)
```

Avec le mode 'a' le contenu du fichier n'est pas effacé ...

Avec le mode 'a' le contenu du fichier n'est pas effacé ...

Avec le mode 'a' le contenu du fichier n'est pas effacé ...

Ne pas oublier de FERMER LE FICHIER (fonction `close()`) avant de quitter le programme, afin de garantir que toutes les modifications ont été réellement enregistrées dans le fichier.

```
[ ]:
```