

Traitement de corpus

Cours n°5 - Manipulation de fichiers

Hee-Soo Choi

UFR Sciences Humaines et Sociales
Université de Lorraine

10 novembre 2021

Introduction

Pour faire du traitement de corpus, vous serez amené à manipuler des fichiers en Python.
Dans ce cours, nous allons voir comment :

- Ouvrir un fichier
- Lire un fichier
- Écrire dans un fichier

1. Ouverture d'un fichier

2. Lecture d'un fichier

3. Écriture d'un fichier

Ouverture d'un fichier

Pour éditer un fichier, il faut d'abord « l'ouvrir ».

On utilise pour cela la fonction `open()`. Elle prend deux paramètres :

- Le chemin du fichier
- Le mode d'ouverture :
 - `r` pour *read*, permet la lecture du fichier.
 - `w` pour *write*, permet d'écrire dans le fichier.
Attention ! À chaque ouverture, le contenu est écrasé.
 - `a` pour *append*, permet d'ajouter du contenu à la fin du fichier.

Pour les modes `w` et `a`, si le fichier n'existe pas, il est créé.

Ouverture d'un fichier

Il existe plusieurs manières d'utiliser la fonction `open()`. Ici, on l'utilisera avec `with open()` pour ne pas à avoir à gérer la fermeture du fichier.

```
1 with open("texte_exemple.txt", "r") as fichier:  
2     [instructions]
```

La variable `fichier` permet de manipuler le fichier.

1. Ouverture d'un fichier

2. Lecture d'un fichier

3. Écriture d'un fichier

Fonction read()

La fonction `read()` renvoie le contenu du fichier dans une chaîne de caractères.

Soit le fichier texte "animaux.txt" avec le contenu suivant :

```
chien  
chat  
lapin  
cheval
```

```
1 with open("animaux.txt", 'r') as fichier:  
2     print(fichier.read())
```

```
1 chien  
2 chat  
3 lapin  
4 cheval
```

Fonction readlines()

La fonction `readlines()` renvoie une **liste** contenant toutes les lignes du fichier.

```
1 with open("animaux.txt", 'r') as fichier:  
2     print(fichier.readlines())
```

```
1 ['chien\n', 'chat\n', 'lapin\n', 'cheval']
```

Attention

`\n` ou `\r\n` représentent un saut de ligne.

C'est un caractère qui indique le passage à la ligne suivante.

Fonction strip()

La fonction strip() supprime les sauts de ligne et les espaces en début et fin de chaîne.

```
1 mot = '      chien      \n'  
2 print(mot.strip())
```

```
1 'chien'
```

Parcourir un fichier

Pour récupérer les lignes d'un fichier, il est plus propre de parcourir le fichier avec une boucle et utiliser la fonction `strip()` sur chaque ligne.

```
1 # On définit une liste vide dans laquelle on va récupérer les lignes
2 liste_lignes = []
3
4 with open("animaux.txt", 'r') as fichier:
5     for ligne in fichier:
6         liste_lignes.append(ligne.strip()) # Ajout dans la liste de chaque
7         ligne sur laquelle la fonction strip() est appliquée
8
9 print(liste_lignes)
```

```
1 ['chien', 'chat', 'lapin', 'cheval']
```

1. Ouverture d'un fichier

2. Lecture d'un fichier

3. Écriture d'un fichier

Fonction write()

La fonction `write()` permet d'écrire dans un fichier.

Reprenons le même fichier "animaux.txt".

```
1 with open("animaux.txt", "w") as fichier:  
2     fichier.write("dauphin")
```

En utilisant le mode d'ouverture "w", le contenu initial du fichier est écrasé.
Il ne contient plus que "dauphin".

Fonction write()

Pour écrire à la suite d'un contenu déjà présent dans un texte, il faut utiliser le mode d'ouverture "a".

```
1 with open("animaux.txt", "a") as fichier:  
2     fichier.write("chat")
```

Le fichier contient :

dauphinchat

La chaîne "chat" est ajoutée directement à la suite. Pour écrire les deux mots l'un après l'autre, il faut ajouter le caractère de saut de ligne à la fin du mot :

```
1 with open("animaux.txt", "w") as fichier:  
2     fichier.write("dauphin\n")  
3     fichier.write("chat\n")
```