

TP 6

Lecture et écriture de fichiers

✂ **Explication** ✂ Ce TP 6 - Lecture et écriture de fichiers est une application directe des notions découvertes dans le Cours 6 - Lecture et écriture de fichiers. Il permet aussi de mettre à jour vos connaissances sur les chaînes de caractères.

Voici les objectifs du TP 6 :

- Savoir lire un fichier et manipuler les données lues,
- Savoir écrire dans un fichier,
- Savoir manipuler les chaînes de caractères et écrire des fonctions.

✖ **ATTENTION !** ✖ Pour le bon fonctionnement du TP, il faut impérativement suivre les consignes suivantes :

1) Copier le dossier *TP6 - Fichiers* dans votre session créer un dossier *TP6 - Fichiers* dans lequel vous copierez les fichiers suivants :

- *TP6 - Document réponse.py*
- *palindromes.txt*
- *citations.txt*
- *messagesecret.txt*
- *names.txt*
- *messagetopsecret.txt*

2) Exécuter *Spyder_reset_FIXE*, puis lancer Spyder en exécutant *Spyder_FIXE*.

Il suffit enfin d'ouvrir le fichier *TP6 - Document réponse.py* qui est le fichier réponse à compléter.

I - Découverte de la lecture et de l'écriture de fichiers

1

Exercice

🕒 Compléter avec soin le tableau suivant :

Rappelons que les lignes en vert sont des explications et que vous n'avez donc pas à les saisir !

Lignes de code à saisir dans l'éditeur	Qu'obtient-on ? Préciser le type des variables
<pre> 1 #pour ouvrir un fichier en lecture 2 mon_fichier=open("citations.txt", 'r') 3 #pour stocker l'intégralité du contenu du fichier 4 contenu=mon_fichier.read() 5 #pour afficher le contenu du fichier 6 print(contenu) 7 #il faut tjrs fermer le fichier 8 mon_fichier.close()</pre>	


 Cliquez ici pour accéder au corrigé

<pre> 1 #ouverture du fichier en mode lecture (non modifiable) 2 mon_fichier=open("citations.txt",'r') 3 4 #lecture de la première ligne 5 contenu=mon_fichier.readline() 6 print(contenu) 7 8 #lecture de la seconde ligne 9 contenu=mon_fichier.readline() 10 print(contenu) 11 12 #lecture des lignes restantes 13 contenu=mon_fichier.readlines() 14 print(contenu) 15 16 mon_fichier.close()</pre>	
<pre> 1 #ouverture du fichier en mode écriture 2 #'w' pour write : écrase le texte existant 3 #'a' pour append : rajoute à la fin du texte existant 4 mon_fichier=open("citations.txt",'w') 5 6 #lecture de la première ligne 7 mon_fichier.write("Ce n'est pas le chemin qui est difficile, mais le difficile qui est le chemin") 8 9 mon_fichier.close()</pre>	

II - Exercices de base

2

Exercice

 Le fichier `names.txt` contient des prénoms séparés par des virgules et non classés par ordre alphabétique. En faisant de la lecture et de l'écriture de fichiers,

- 1) Quel est le nombre de prénoms qu'il contient ?
- 2) Quelle commande faut-il taper pour récupérer le dernier prénom de ce fichier ?
- 3) Créer un fichier `namesalpha.txt` contenant tous ces prénoms classés par ordre alphabétique et dans lequel vous aurez rajouté le vôtre.
- 4) Quel est le rang de votre prénom ?

3

Exercice



- 1) Écrire une fonction

`occurrence(lettre,mot)`

qui prend en paramètres une chaîne de caractères `lettre` et **renvoie** son nombre d'occurrences dans une autre chaîne de caractères `mot` donnée.

```

1 # voici un exemple d'appel :
2 >>> occurrence("E","ELEPHANT")
3 2
```

- 2) Combien de voyelles contient le fichier `names.txt` ? et de consonnes ?

4**Exercice**

🕒🕒 Un palindrome de lettres est un mot qui peut se lire aussi bien de gauche à droite que de droite à gauche.

Exemple Le mot « radar » est un palindrome de lettres !

1) Écrire une fonction

palindrome(mot),

qui prend en paramètre une chaîne de caractères mot et teste si elle représente un palindrome ou pas.

```
1 # voici deux exemples d'appel :
2 >>> palindrome("radar")
3 True
4
5 # voici deux exemples d'appel :
6 >>> palindrome("hello")
7 False
```

2) En faisant de la lecture et de l'écriture de fichiers,

- Déterminer le nombre de mots à tester dans le fichier `palindromes.txt`.
- Combien de palindromes possède-t-il ? Quels sont-ils ?

III - Pour aller plus loin

5**Exercice**

🕒🕒🕒 **(Chiffrement César)** César envoyait des messages codés à ses généraux en décalant de 3 lettres. Cette façon de coder s'appelle désormais le *Chiffrement César* et l'entier 3 désigne la clé de ce chiffrement.

<i>Alphabet clair</i>	a	b	y	z
<i>Alphabet crypté (avec une clé de 3)</i>	d	e	b	c

Exemple Ainsi, le mot « bonjour » devient « erqmrxu ».

Signalons qu'on ne gère ni les accents ni les majuscules par la suite et qu'on fera nécessairement de la lecture et de l'écriture de fichiers lorsqu'il y a des fichiers à manipuler.

1) Écrire une fonction

code(mot,n),

qui prend en paramètres une chaîne de caractères composée de lettres (en minuscules et sans accents) et d'un entier n, et renvoie la chaîne de caractères où toutes les lettres ont été décalées de l'entier n.

```
1 >>> code("bonjour",3)
2 "erqmrxu"
```

- Après avoir vaincu le chef de l'armée adverse, César décide de lui faire parvenir un message crypté avec la clé 17. Ce message crypté se trouve dans le fichier `messagesecret.txt`. Qu'a donc bien pu dire César au chef de l'armée adverse après l'avoir vaincu ?
- Le fichier `messagetopsecret.txt` contient un message top secret à décoder. La clé du chiffrement n'est pas connue mais on sait que la lettre i dans le message clair est la lettre la plus fréquente.
 - Déterminer la clé du chiffrement.
 - Créer un fichier `messagetopsecret_traduit.txt` dans lequel figurera le message décodé.