
IngeSUP - TD 06 - I/O fichiers txt

“Cookie : Anciennement petit gâteau sucré, qu'on acceptait avec plaisir. Aujourd'hui : petit fichier informatique drôlement salé, qu'il faut refuser avec véhémence.”

Luc Fayard

Exercice 06.1

1. Créez le fichier "exo1.txt" en l'ouvrant en mode écriture, écrivez un caractère dedans, puis refermez-le.

Entrée []:



VOTRE CODE ICI

2. Ouvrez-le en mode lecture, affichez ce que vous lisez dedans, puis refermez-le.

Entrée []:



VOTRE CODE ICI

3. Ouvrez-le en mode écriture, écrivez "Hello " dedans, puis refermez-le.

Entrée []:



VOTRE CODE ICI

4. Ouvrez-le en mode lecture, affichez ce que vous lisez dedans, puis refermez-le. Que remarquez-vous ?

Entrée []:



VOTRE CODE ICI

Entrez votre réponse ici

5. Ouvrez-le en mode ajout, écrivez "World !" dedans, puis refermez-le.

Entrée []:



```
# VOTRE CODE ICI
```

6. Ouvrez-le en mode lecture, affichez ce que vous lisez dedans, puis refermez-le. Que remarquez-vous ?

Entrée []:



```
# VOTRE CODE ICI
```

Entrez votre réponse ici

7. Ouvrez-le en mode ajout, écrivez en sautant une ligne "Bonjour tout le monde" dedans, puis refermez-le.

Entrée []:



```
# VOTRE CODE ICI
```

8. Ouvrez le sans utiliser de mode, affichez le contenu de chaque ligne, puis refermez-le. Que remarquez-vous ?

Entrée []:



```
# VOTRE CODE ICI
```

Entrez votre réponse ici

Exercice 06.2

1. Que fait le code Python suivant ?

```
compteur = 0
f = open("test.txt", "r")
caractere = f.read(1)
while caractere != '':
    if caractere == "a":
        compteur += 1
```

```
caractere = f.read(1)

f.close()
print("Résultat: ", compteur)
```

Entrez votre réponse ici

2. Pourquoi à t-on mis cette commande à l'intérieur du while ?

```
caractere = f.read(1)
```

Entrez votre réponse ici

3. Essayez de deviner l'affichage des suites d'instructions suivantes, puis testez dans la console python (justifiez votre réponse) :

```
f = open("fichier.ext", "w")
f.write("bonjour")
f.close()
f = open("fichier.ext", "r")
f.read(3)
a = f.read(4)
f.close()
print(a)
```

Entrez votre réponse ici

4. Essayez de deviner l'affichage des suites d'instructions suivantes, puis testez dans la console python (justifiez votre réponse) :

```
f = open("test.ext", "w")
f.write("une ligne\n") # \n sert à sauter la ligne..
f.write("Une autre ligne\n")
f.write("unE dernière ligne\n")
f.write("en fait non\n")
f.close()

f = open("test.ext", "r+") # Mode Lecture/ecriture
f.readline()
f.read(5)
f.readline()
a = f.read(3)
print(a)
f.close()
```

Entrez votre réponse ici

Exercice 06.3

Cet exercice traite un vrai listing des candidats du concours Mines-Ponts 2013, filère MP, récupéré sur Internet.

Chaque ligne est constituée de quatre champs : numéro de candidat, Nom et Prénom, Admissible ou refusé, numéro de série d'oral si admissible.

1. Téléchargez le fichier "admissibles.txt" que vous trouverez sur Moodle. Copiez-le dans le même répertoire que votre notebook. Ouvrez avec un éditeur de textes le fichier admissibles.txt. Quel semble être le caractère qui sépare les quatre champs ?

Entrez votre réponse ici

2. Complétez la première ligne du script suivant pour ouvrir le fichier en mode lecture, votre fichier admissibles.txt:

```
f = open(...  
a= f.readline() # La première ligne du fichier  
b= f.readline() # La deuxième ligne du fichier  
f.close()
```

Exécutez le script, puis affichez `a` .

Entrée []:



VOTRE CODE ICI

3. Quel est donc le caractère qui sépare les quatre colonnes ?

Entrez votre réponse ici

4. Ecrivez les instructions qui permettent de connaître le nombre de candidats qui correspond au nombre de lignes du fichier.

Entrée []:



VOTRE CODE ICI

5. Appliquez la méthode `split()` à une ligne du fichier en utilisant comme paramètre le séparateur trouvé dans la question (3) en complétant puis en exécutant les lignes du script suivant :

```
f = open( ....  
a= f.readline() # La première ligne du fichier  
l=a.split(...  
print(l)  
f.close()
```

Que retourne la méthode `.split()` utilisée avec le séparateur en paramètre ?

Entrée []:



VOTRE CODE ICI

Entrez votre réponse ici

6. Déterminez le nombre d'admissibles au concours.

Entrée []:



VOTRE CODE ICI

7. Écrivez un fichier "noms_admissibles.txt" contenant tous les noms et prénoms des candidats admissibles.

Entrée []:



VOTRE CODE ICI

Exercice 06.4

1. Créez un sous-répertoire TablesMultiplication ;
2. Dans ce sous-répertoire, créez les fichiers textes `table_de_x.txt` où `x` est un entier compris entre 1 et 100. Chaque fichier `table_de_x.txt` sera composé de 10 lignes de texte:

`0 x 0 = 0`

`0 x 1 = 0`

`0 x 2 = 0`

`...`

Entrée []:



VOTRE CODE ICI

Exercice 06.5

Ecrivez un programme qui :

1. écrit dans un fichier texte `notes_etudiants.txt` les 10 premières valeurs réelles comprises entre 0 et 20 entrées par l'utilisateur.

Entrée []:



```
# VOTRE CODE ICI
```

2. écrit dans un autre fichier texte `moyenne.txt` la moyenne des notes lues dans le fichier `notes_etudiants.txt`.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 06.6

On se propose de réaliser dans cette exercice un générateur aléatoire de noms - prénoms.

Ce générateur se base sur une liste des noms les plus portés en France, ainsi que sur les prénoms les plus donnés en France. Téléchargez le fichier "FichierNoms.txt", "FichierPrenomsFilles", "FichierPrenomsGarçons.txt" que vous trouverez sur Moodle. Ecrivez un programme qui :

- demande à l'utilisateur combien de noms-prénoms de garçons doit-il générer et également combien noms-prénoms de filles ;
- crée un fichier texte où chaque ligne du fichier contient : un nom tiré aléatoirement parmi les noms les plus portés, un prénom tiré aléatoirement parmi les prénoms les plus donnés ainsi que le sexe, représenté par les lettres F ou M selon que le prénom donné soit un prénom de filles ou de garçon.

Entrée []:



```
# VOTRE CODE ICI
```

Corrigé du TD 06

Vous pouvez retrouver le corrigé de ce TD [ici \(Corrig%C3%A9s/Corrig%C3%A9_TD%2006.ipynb\)](#).