

PYTHON Lire et écrire un fichier texte

1) Introduction

Pour garder des résultats donnés par un programme, pour coder des documents importants, il est indispensable de savoir manipuler des fichiers avec PYTHON.

Dans un premier temps, nous allons manipuler des fichiers ne contenant que du texte (chaînes de caractères).

2) Exemple

Tapez et exécutez le programme suivant:

```
suite=[1,1]
n=30
for k in range(n):
    suivant=suite[k]+suite[k+1]
    suite.append(suivant)
print(suite)
```

Vous avez compris que ce programme génère les premiers termes de la célèbre suite de Fibonacci!

Imaginons qu'un ami, qui n'y connaît rien en programmation, vous demande de lui envoyer par courriel la liste des 300 premiers termes. (ou bien plus!!)

Votre mission, et vous êtes obligés de l'accepter, est de lui répondre, sans trop vous fatiguer.

3) Instructions

L'instruction basique qui crée un fichier dans le même répertoire que celui qui contient le programme (il est possible de le créer ailleurs,....voir la doc!) est:

```
fichier=open("nom",'w')
```

fichier est le nom de votre fichier dans le programme python

"nom" est le nom de votre fichier de sortie

w signifie "write", on ouvre le fichier pour écrire dedans.

Attention, si un fichier du même nom existe déjà dans le répertoire il sera écrasé sans aucune alerte!!!!!!

Pour écrire les infos dans le fichier, c'est simple:

```
fichier.write("info")
```

Et on finit toujours (que ce soit en lecture ou écriture) par:

```
fichier.close()
```

Complétez le programme précédent en ajoutant ces 4 lignes:

```
fichier=open("fibonacci.txt",'w')
for terme in suite:
    fichier.write(str(terme))
fichier.close()
```

Exécutez le programme, ouvrez le fichier "fibonacci.txt" et constatez le carnage!

Python a écrit tous les nombres à la suite (ce qui servira un jour....).

Il faut ajouter après chaque nombre le "caractère" qui produit un passage à la ligne: "\n".

```
fichier=open("fibonacci.txt",'w')
for terme in suite:
    fichier.write(str(terme)+"\n")
fichier.close()
```

4) Exercice

Vous allez créer pour votre ami un beau fichier.

D'abord, il faut pouvoir choisir le nombre de termes de la suite de Fibonacci, ainsi que le prénom de votre ami.

Et puis mettre quelques mots dans votre texte!

Votre fichier texte devra ressembler à:

```
Bonjour mon cher Archibal,  
voici la liste des 300 termes demandés:
```

```
u(0)=1
```

```
u(1)=1
```

```
.
```

```
.
```

```
u(299) = 222232244629420445529739893461909967206666939096499764990979600
```

```
Ce fut un plaisir .....
```

De plus votre fichier devra être nommé "Archibal300.txt". (dépend donc du destinataire et du nombre)

Supplément: vous pouvez indiquer en bas du fichier, le temps mis par python pour effectuer le travail!

5) Lecture d'un fichier

Exemple: on veut savoir combien de "A" contient le livre "Germinal" d'Emile Zola.

(L'œuvre est dans le domaine public, il n'y a pas de problème pour la trouver sur le net)

Allez chercher sur www.isn.cligniez.fr, dans les ressources, le fichier *germinal01.txt*.

Placez le dans le répertoire qui contient vos programmes python.

C'est un fichier où toutes les lettres ont été converties en majuscules sans accents.

On va l'ouvrir avec python et compter les lettres.

fichier=open("germinal01.txt",'r') On ouvre le fichier en lecture r = read

Il y a 2 méthodes principales de lecture, *readline* et *readlines*

readline lit une ligne à la fois et renvoie une chaîne

readlines lit toutes les lignes et renvoie une liste de chaînes.

texte=fichier.readlines() crée une liste appelée texte qui contient chaque ligne du fichier

texte=str(texte) convertit la liste en une seule chaîne.

Dans python on dispose d'un très grand nombre de méthodes pour les chaînes!!!

Ici on va utiliser **count("mot")**, qui compte le nombre d'occurrences de "mot" dans la chaîne.

Et voici le programme:

```
fichier=open("germinal01.txt", 'r')  
texte=fichier.readlines()  
fichier.close()  
texte=str(texte)  
nombre=texte.count("A")  
print("Germinal contient ",nombre," fois la lettre A")
```

6) Exercice

Vous aller ouvrir Germinal, compter pour chaque lettre le nombre d'occurrences.

Evidemment, vous n'allez pas écrire 26 lignes, mais plutôt une boucle for, en utilisant les fonctions **ord** et **chr** qui gèrent le code ASCII des caractères. `ord("A") = 65` et `chr(65) = "A".....`

En sortie vous écrirez votre rapport dans un fichier texte: rapport_germinal.txt, dont voici un aperçu:

```
Analyse de Germinal: j'ai trouvé 786296 lettres
```

```
Nombre de A = 72409
```

```
Nombre de B = 8162
```

```
.....
```

```
J'ai effectué le travail en 6.867573113342618 secondes, même pas peur!
```