

Manipulations de fichiers Lire et écrire dans un fichier

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

Répertoire courant

Les exercices suivants vont écrire des fichiers dans le répertoire courant. Auparavant, vous devez vérifier dans quel répertoire vous êtes positionné. La fonction `getcwd()` du module `os` permet d'afficher le répertoire courant :

```
>>> from os import getcwd, chdir
>>> getcwd()
```

La fonction `chdir()` du module `os` permet de modifier ce répertoire courant. Modifiez le répertoire courant de manière à ce qu'il pointe sur le votre répertoire de travail. Ainsi les fichiers que vous allez créer s'enregistreront au même endroit que vos programmes python.

Exercice 1 : Changer de répertoire courant pour vous positionner dans un répertoire de votre espace personnel.

Instructions d'écriture/de création

Exercice 2 : Construire (ou retrouver) une fonction **Fibonacci()** dépendant d'un entier n et renvoyant le terme d'indice n de la suite de Fibonacci.

L'instruction basique qui crée un fichier dans le même répertoire que celui qui contient le programme :

```
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 du texte dans le fichier, c'est simple :

```
fichier.write("... le texte est ici ...")
```

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

```
fichier.close()
```

Exercice 3 : Exécuter le programme suivant :

```
fichier=open("fibonacci.txt",'w')
for i in range(0,1000):
    fichier.write(fibonacci(i))
fichier.close()
```

Corriger l'erreur produite et ouvrez par le *bloc note* de windows (ou Notepad) le fichier créé et, après avoir compris la construction, modifier le programme afin de visualiser par lignes les premiers termes de la suite de Fibonacci.

Exercice 4

Créer un fichier respectant les contraintes suivantes : D'abord, il faut pouvoir choisir le nombre de termes de la suite de Fibonacci, ainsi que le prénom de la personne utilisant le programme.

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
```

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!

Pour ce faire, Python dispose d'une bibliothèque `time` contenant l'instruction `time()` qui retourne un float qui représente le temps écoulé (en sec) depuis le 1er Janvier 1970. Par différence de temps, vous aurez la durée recherchée.

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)

germinal01.txt 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 pour 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.
```

Attention, bien ouvrir le fichier avec la méthode 'r' pour ne pas supprimer le fichier du serveur !

Exercice 5

Vous aller ouvrir Germinal,

1) Dans un premier temps, compter le nombre de 'A'

2) Puis compter chaque lettre et le nombre d'occurrences pour chacune d'entre-elles.

Vous pouvez utiliser les fonctions ord et chr qui gèrent le code ASCII des caractères.

```
ord("A") = 65 et chr(65) = "A".....
```

En sortie vous écrivez 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
```

```
.....
```

```
Le travail a été effectué en 6.867573113342618 secondes.
```

Modification d'un fichier

Il existe un troisième mode d'ouverture de fichier :

```
"a" (append) indique un mode en ajout.
```

Une méthode importante sur les chaînes

Pour extraire des données dans une ligne, on peut utiliser la méthode split. Par exemple, f=open('exemple.txt','r') où exemple.txt contient la seule ligne suivante "La France est un beau pays, même si les Français sont râleurs."

En tapant f.split(',') : on coupe la chaîne selon le caractère ',' et une liste de chaînes est rendue. On peut utiliser de même les méthodes f.split(':') pour séparer selon le symbole ':' ou encore f.split('\t') selon les tabulations.

Cette méthode est très importante pour lire des fichiers de données du type csv (le type csv est à celui d'excel xls ce que le txt est au doc du word) fournis par des tableurs.

Exercice 6

Vérifier que le texte "La disparition" de Georges Perec ne contient pas de "e".

Exercice 7 Analyse d'un fichier de données de type txt

Vous disposez d'un fichier texte contenant les millièmes premières décimales de π . Il s'agit de déterminer l'ordre des fréquences d'apparition de chacun des dix chiffres. Une rapide recherche d'un tri, comme le **tri-bulle** devrait vous aider au classement.

Exercice 8 Statistiques sur le nombre de visites du site internet du lycée dans un fichier de type csv

Vous disposez dans un fichier du nombre de visites sur le site internet de votre lycée depuis le 12 mars 2016. Après avoir chargé ce fichier, il s'agit d'établir le classement du nombre de visites de ce site et d'indiquer les 10 jours de l'année ayant le plus de visites.

```

from os import chdir
def Fibonacci(n):      # EXERCICE 2
    if n==0:
        return 1
    elif n==1:
        return 1
    else:
        u=1
        v=1
        for k in range(2,n+1):
            w=u+v
            u=v
            v=w
        return w

chdir('C:\\Users\\Maison\\Desktop\\')    # EXERCICE 1
fichier=open("fibonacci.txt",'w')
for i in range(0,1000):
    fichier.write(str(Fibonacci(i)) )    #ERREUR DE L'EXERCICE 3
fichier.close()

n=int(input("Combien de termes de la suite souhaitez-vous ?"))    # EXERCICE 4
prenom=input('Quel est votre prénom ?')
fichier=open(prenom+str(n)+".txt",'w')
for i in range(0,n):
    fichier.write(str(fibonacci(i))+'\n')
fichier.close()

fichier=open("germinal01.txt", 'r') #EXERCICE 5
texte=fichier.readlines()
fichier.close()
texte=str(texte)
nombre=texte.count("A")
print("Germinal contient ",nombre," fois la lettre A")

from time import time
temps_debut=time()
fichier=open("germinal01.txt", 'r')
texte=str(fichier.readlines())
fichier.close()
L=[0 for i in range(0,26)]
n=0
for i in range(0,len(texte)):
    k=ord(texte[i])
    if 65<=k<=65+25:
        L[k-65]=L[k-65]+1
        n=n+1
temps_fin=time()
fichier=open("rapport_germinal.txt",'w')
fichier.write("Analyse de Germinal : j'ai trouvé "+str(n)+" lettres.\n")
for i in range(65,65+25):
    fichier.write("Nombre de "+chr(i)+" = "+str(L[i-65])+"\n")
fichier.write('Le travail a été effectué en '+str(temps_fin-temps_debut)+" secondes.")
fichier.close()

```

```

fichier='disparition.txt'
clair=open(fichier,'r') #en lecture seule#
texte=clair.read()
#print(texte)
clair.close()
c=0
N=len(texte)
for caractere in texte:
    if caractere=='e' or caractere=='E' or caractere=='é' or caractere=='è' or caractere=='ê':
        c=c+1
print(c) #on doit trouver 0 logiquement...

```

```

def compter(c):      # EXERCICE 7
    T=[[0]*10]+[[0]*10]
    for i in range(0,10):
        T[0][i]=i
    for i in range(0,len(c)):
        if c[i]!='\n':
            x=int(c[i])
            T[1][x]=T[1][x]+1
    return(T)

```

```

def tri_bulle_2(t):
    n=len(t[0])
    echange_effectue=True
    while echange_effectue==True:
        echange_effectue=False
        for i in range(0,n-1):
            if t[1][i]>t[1][i+1]:
                t[0][i],t[0][i+1]=t[0][i+1],t[0][i]
                t[1][i],t[1][i+1]=t[1][i+1],t[1][i]
            echange_effectue=True
    return t

```

```

F=open("C:\\Users\\Maison\\Desktop\\pi1000000.txt")
c=F.read()
tableau=compter1(c)
print(tableau)
tableau=tri_bulle_2(tableau)
print(tableau)

```

```

def ouvrir_fichier():      #EXERCICE 8
    F=open('statistiques.csv')#Pour aller chercher le fichier contenant les données.
    j=F.readlines()#Lit le fichier.
    j.pop(0) #Supprime le premier élément de la liste car il ne nous intéresse pas.
    T=[]#Créer une liste T, vide.
    for i in range(0,len(j)): #séparation chaque binôme date/effectif de la liste en deux éléments et les met dans la
        liste T.
        a=j[i]
        T.append(a.split(", "))
    for i in range(0,len(T)):#considération des effectifs comme des entiers.
        x=T[i][1]
        T[i][1]=int(x[0:len(x)-1])
    return(T)

```

```

def tri(T):
    j = []#Créer une liste j, vide qui va contenir la liste initiale triées par ordre décroissant.
    for x in range(0,len(T)):#Boucle permettant de trier la liste T par ordre décroissant d'effectifs
        c=0
        for i in range(1,len(T)):
            if T[i][1]>T[c][1]:
                c=i
        j.append(T[c])#Ajoute à la liste j chaque élément de T par ordre décroissant.
        T.pop(c)#Enlève de T l'élément que l'on a mit dans j.
    return(j)

def pg():
    H=ouvrir_fichier()
    j=tri(H)
    top10=[] #créer une liste vide qui va contenir le top 10.
    for i in range(0,10): #Boucle qui prend les 10 premiers éléments de j pour les mettre dans top10.
        top10.append(j[i])
    print("Voici les 10 meilleures fréquentations du site:") #Présentation des résultats.
    for i in range(0,10): #Boucle qui énumère les dix meilleures performances.
        print("Le",(top10[i][0]),", il y a eu",(top10[i][1]),"visiteurs sur le site du lycée.")

pg()

```