

Apprentissage de Python

■ ■ ■ Manipulation de fichiers et de listes

1 – Demander le nom du fichier à ouvrir :

```
#!/usr/bin/python3

import sys

nom_fichier = input("Nom fichier:")

try:
    descripteur_fichier = open(nom_fichier, "r")
except Exception as e:
    print(e.args)
    sys.exit(1)

compteur_lignes = 0
while 1:
    ligne = descripteur_fichier.readline()
    if not ligne:
        break
    compteur_lignes += 1
print("Nombre de lignes :", compteur_lignes)

descripteur_fichier.close()
```

2 – Numéro de sécurité sociale :

```
#!/usr/bin/python3

import sys

numéro_ss = input("Numéro de sécurité sociale :")
clé_ss = input("Clé :")

if (len(numéro_ss) != 13):
    print("Erreur de saisie du numéro de sécurité sociale")
    sys.exit(1)

numéro_ss_numérique = int(numéro_ss)
clé_ss_numérique = int(clé_ss)

if ((97 - (numéro_ss_numérique % 97)) != clé_ss_numérique):
    print("Le numéro de sécurité sociale et la clé ne correspondent pas")
    sys.exit(1)

print("Sexe :\t", numéro_ss[0])
print("Année:\t\t", numéro_ss[1:3], "\tMois:\t", numéro_ss[3:5], "\tDpt:\t", numéro_ss[5:7])
print("Commune:\t", numéro_ss[7:10], "\tNuméro:\t", numéro_ss[10:])
```

3 – Écrire un programme réalisant la copie d'un fichier en inversant les lignes de celui-ci.

```
#!/usr/bin/python3

nom_fichier = input("Donnez le nom du fichier :")

try:
    entrée = open(nom_fichier, "r")
    sortie = open(nom_fichier+'.inverse', "w")

except Exception as e:
    print(e.args)

lignes = entrée.readlines()
entrée.close()
lignes.reverse()
for l in lignes:
    sortie.write(l)
sortie.close()
```

4 – écrire un programme prenant la liste des fichiers contenus dans un répertoire, et qui ouvre et affiche la première ligne de chacun de ces fichiers :

```
#!/usr/bin/python3

import sys
import subprocess

resultat = subprocess.run('ls *.py', shell=True, stdout=subprocess.PIPE)
liste_fichiers = resultat.stdout.splitlines()

for nom_fichier in liste_fichiers:
    try:
        f = open(nom_fichier, "r")
    except Exception as e:
        continue
    ligne = f.readline()
    print(ligne)
    f.close()
```

■ ■ ■ Représentation hexadécimale

5 – Écrire un programme qui reproduit le traitement de la commande « xxd » du shell :

```
#!/usr/bin/python3
import sys

nom_fichier = 'xxd.py'

try:
    f = open(nom_fichier, 'rb')
except Exception as e:
    print(e.args)
    sys.exit(1)

décalage = 0
ligne = b''
while 1:
    if len(ligne) == 0:
        print('{0:0{1}X}'.format(décalage, 4), ': ', end='')
        car = f.read(1)
        if not car:
            break
        if (ord(car)<128) and (ord(car)>31):
            ligne += car
        else :
            ligne += b'.'
        print('{0:0{1}X} '.format(ord(car), 2), end='')
        if len(ligne) == 16:
            print(' ', str(ligne, encoding='UTF-8'))
            ligne = b''
            décalage += 1
    if len(ligne):
        print(' '* (16-len(ligne)), end='')
        print(' ', str(ligne, encoding='UTF-8'))

f.close()
```

6 – Écrire un programme de chiffrement utilisant la technique de l'OTP, « One Time Password » :

```
#!/usr/bin/python3

clé=b"CLESUPERSECRETE"
message=b"message a chiffrer"

def otp(m,c):
    message_entiers = list(m)
    clé_entiers = list(c)
    chiffre = b""
    for i in range(0,len(message_entiers)):
        chiffre += bytes([message_entiers[i]^clé_entiers[i%len(clé)]])
    return chiffre

print("Message :",message.hex())
print("Clé :",clé.hex())
chiffré = otp(message,clé)
print (chiffré.hex())
déchiffré = otp(chiffré,clé)
print (déchiffré.hex())
```