

# Chapitre : Les Fichiers en Python

Ph. Hunel

## Fichier, pour faire quoi ?

- ▶ Objet représentant un emplacement sur le disque dur dans lequel des données sont conservées
- ▶ Stockage de tout type de données, y compris du texte, des nombres, des images, des vidéos, etc.
- ▶ Deux types de fichiers en Python :
  - ▶ **Fichiers texte**
    - ▶ contiennent du texte brut, tel que du code Python, des données brutes, ou tout autre type de texte.
    - ▶ ouverts généralement en mode lecture ou écriture.
  - ▶ **Fichiers binaires**
    - ▶ contiennent des données binaires, telles que des images, des vidéos, ou des fichiers exécutables.
    - ▶ ouverts généralement en mode binaire.

## La bibliothèque **os**

- ▶ Manipulation de fichiers (et de répertoires)
  1. parcourir
  2. renommer
  3. création/destruction de répertoires
- ▶ Pas de copie de fichiers

## La bibliothèque **os**

- ▶ `getcwd()` : fournit le répertoire courant
- ▶ `chdir(nomrep)` : se place dans le répertoire `nomrep`
- ▶ `rename(src,dest)` : renomme `src` en `dest`
- ▶ `remove(chemin)`, `rmdir(chemin)` : supprime `chemin`
- ▶ `mkdir(chemin)` : crée le répertoire `chemin`
- ▶ `listdir(chemin)` : liste des fichiers/rep de `chemin`
- ▶ `system(commande)` : exécute la commande `commande`

## La bibliothèque **os**

► Exemple d'utilisation:

```
1 import os
2 rep_actuel = os.getcwd()
3 print(rep_actuel) # renvoie le répertoire actuel
```



## La bibliothèque **os**

► Exemple d'utilisation:

```
1 import os
2 rep_actuel = os.getcwd()
3 print(rep_actuel) # renvoie le répertoire actuel
4 os.chdir("/Users/philippehunel/Library/Cours/Python")
5 rep_actuel = os.getcwd()
6 print(rep_actuel) # renvoie le répertoire actuel
7 contenu = os.listdir(rep_actuel)
8 for element in contenu:
9     print(element)
```

## La bibliothèque **os.path**

7

- ▶ Manipulation courante des chemins

- ▶ Utilisation :

```
from os.path import *
```

- ▶ OU

```
import os.path
```

## La bibliothèque **os.path**

8

- ▶ `split(chemin)` : fournit le tuple (`repertoire, fichier`)
- ▶ `join(chemin, ...)` : fournit un nom complet :`/plus/de/un/rep/unFichier`
- ▶ `exists(chemin)` : vrai si chemin existe
- ▶ `isfile(chemin)` : vrai si chemin est un fichier
- ▶ `isdir(chemin)` : vrai si chemin est un répertoire
- ▶ `walk(chemin, fonc, arg)`

## La bibliothèque `os.path`

9

## ► Exemple d'utilisation:

```
1 import os.path
2 rep_actuel = os.getcwd()
3 print("#####\n",rep_actuel) # renvoie le répertoire actuel
4
5 nom_complet=os.path.join(rep_actuel,"bidon.py")
6 print(nom_complet)
7 (repertoire, fichier)=os.path.split(nom_complet)
8 print("Le dossier : \n",repertoire)
9 print("Le fichier : ",fichier)
```

```
macOS  
#####  
/Users/philippehunel  
/Users/philippehunel/bidon.py  
Le dossier :  
/Users/philippehunel  
Le fichier : bidon.py
```

```
Windows
#####
C:\Users\phili
C:\Users\phili\bidon.py
Le dossier :
C:\Users\phili
Le fichier : bidon.py
```

9

## La bibliothèque **shutil**

10

- ▶ Fournit des fonctions de copie
  - ▶ `copyfile(src, dest)`
    - ▶ copie du fichier sans les droits
    - ▶ copie du fichier `src` en fichier `dest`
    - ▶ les chemins peuvent être absous ou relatifs.
  - ▶ `copy(src, dest)`
    - ▶ comme `copyfile`
    - ▶ contrairement à `copyfile`, permet de copier un fichier dans un dossier
    - ▶ copie complète du fichier (y compris les droits)

## Fichiers logiques

- ▶ Pour manipuler le contenu de fichier il faut définir des **fichiers logiques** (ou **objets fichier**)
- ▶ Deux fonction centrales :
  - ▶ `open(nom, mode)` : ouvre le fichier `nom` avec le mode `mode`
  - ▶ `obFic.close()` : ferme l'objet fichier `obFic`
- ▶ Exemple :
 

```
▶ ficLog=open('test.txt','r') # Ouvre ficLog
▶ ficLog.close() # Ferme ficLog
```

## Création, ouverture de fichier

- ▶ Création d'un fichier avec `open()`
- ▶ deux arguments : le nom du fichier et le mode
- ▶ Le mode d'ouverture définit comment le fichier sera ouvert
- ▶ modes d'ouverture les plus courants :
  - ▶ "r" :
  - ▶ Mode lecture. Le fichier est ouvert en mode lecture seule.
  - ▶ "w" :
  - ▶ Mode écriture. Le fichier est ouvert en mode écriture seule.
  - ▶ "a" :
  - ▶ Mode ajout. Le fichier est ouvert en mode ajout. Les données sont ajoutées à la fin du fichier.

## Création, ouverture de fichier

- ▶ Exemple :

```

1   # Ouvre un fichier texte en mode lecture
2   with open("mon_fichier.txt", "r") as f:
3       # Lit le contenu du fichier
4       contenu = f.read()
5
6   print(contenu)

```

## Lecture

- ▶ Soit `ficLog=open('entree.txt','r')`
- ▶ `ficLog.read()` : fournit la totalité de `ficLog` sous forme d'une chaîne de caractères
- ▶ `ficLog.read(n)` : fournit n caractères de `ficLog` à partir de la position courante
- ▶ `ficLog.readline()` : fournit une ligne de `ficLog` sous forme d'une chaîne de caractères
- ▶ `ficLog.readlines()` : fournit la totalité de `ficLog` sous forme d'une liste de chaîne de caractères

## Écriture

- ▶ Soit `ficLog=open('sortie.txt', 'w')`
- ▶ `ficLog.write(ch)` : écrit la chaîne `ch` dans `ficLog`
- ▶ `ficLog.writelines(list)` : écrit les chaînes de `list` dans `ficLog`

## Ecriture

- ▶ Exemple :

```

1 # Ouvre un fichier texte en mode écriture
2 with open("mon_fichier.txt", "w") as f:
3     # Écrit une chaîne de caractères dans le fichier
4     f.write("Ceci est une chaîne de caractères.")

```

## Fermeture des fichiers

- ▶ important de fermer les fichiers après utilisation.
- ▶ permet de libérer les ressources utilisées par le fichier.
- ▶ méthode `close()`

## Fermeture de fichier

- ▶ Exemple :

```
1 # Ouvre un fichier texte en mode lecture
2 with open("mon_fichier.txt", "r") as f:
3     # Lit le contenu du fichier
4     contenu = f.read()
5
6     # Ferme le fichier
7     f.close()
```

## Exemple

- ▶ Programme de recopie du contenu d'un fichier dans un autre

```

fic1=open('source.txt','r') # ouverture des fichiers
fic2=open('but.txt','w')
ch=fic1.readline()
while ch!="":
    fic2.write(ch)
    ch=fic1.readline()
fic1.close() # fermeture des fichiers
fic2.close()

```

## Exemple

- ▶ écrire un programme qui recopie le fichier data.txt dans le fichier data.txt en supprimant toutes les lignes qui commencent par le symbole #.