Python : Gestion de fichiers

In [23]:
```
import os
from pathlib import Path # !!! ===>  python >= 3.4
```

1- Gestion des chemins de fichiers avec pathlib

- module os : créer fichier/dossier, concaténation de plusieurs chemins
- module shutil : déplacer des fichiers
- module glob = scanne le disque dur pour récupérer des chemins de fichiers

Récupérer le dossier parent

`os.path.dirname("/Users/Doc/test.py")`

= renvoi le chemin : `"/Users/Doc"`

Récupérer l'extension d'un fichier :

In [9]:
```
os.path.splitext("/Users/Doc/test.py")[1]
```

Out[9]:
```
'.py'
```

Pathlib

Idem mais avec Pathlib

In [15]:
```
path = Path("/Users/Doc/test.py")
print("dossier parent :", path.parent)
print("extension du fichier :", path.suffix)
```

```
dossier parent : \Users\Doc
extension du fichier : .py
```

Récupérer le dossier utilisateur

In [19]:
```
Path.home()
```

Out[19]:
```
WindowsPath('C:/Users/utilisateur')
```

Récupérer le dossier courant

In [25]:
```
Path.cwd()
```

Out[25]:
```
WindowsPath('C:/Users/utilisateur/Desktop/Projet_CDE/code/BDD/Libraires')
```

Création d'un chemin

In [21]:
```
p = Path("/Users/Bob/mon_chemin")
p
```

Out[21]:
```
WindowsPath('/Users/Bob/mon_chemin')
```

Récupérer le dossier parent

In [28]:
```
p.parent
```

Out[28]:
```
WindowsPath('/Users/Bob')
```

Concaténation des chemins

In [29]:
```
# 1er chemin
p = Path.home()
p
```

Out[29]:
```
WindowsPath('C:/Users/utilisateur')
```

In [30]:
```
# concaténation
p / "Documents"
```

Out[30]:
```
WindowsPath('C:/Users/utilisateur/Documents')
```

In [32]:
```
# concaténation
p / "Documents" / "test.py"
```

Out[32]:
```
WindowsPath('C:/Users/utilisateur/Documents/test.py')
```

Autre méthode : `joinpath`

In [34]:
```
p.joinpath("Musique", "son.mp4")
```

Out[34]:
```
WindowsPath('C:/Users/utilisateur/Musique/son.mp4')
```

Récupérer l'extension :

In [35]:
```
p.joinpath("Musique", "son.mp4").suffix
```

Out[35]:
```
'.mp4'
```

In [36]:
```
(p / "Documents" / "main.py").suffix
```

Out[36]:
```
'.py'
```

Récupérer des informations sur un chemin

In [63]:
```
p = Path("/User/Documents/Mon_dossier/index.html")
```

Nom du fichier :

In [64]:
```
p.name
```

Out[64]:
```
'index.html'
```

Nom du fichier (sans l'extension)

In [65]:
```
p.stem
```

Out[65]:
```
'index'
```

Extension du fichier

In [66]:
```
p.suffix
# suffixes si plusieurs extensions, ex : 'indx.tar.gz' pour une archive
```

Out[66]:
```
'.html'
```

Récupérer toutes les parties du chemin

In [67]:
```
p.parts
```

Out[67]:
```
('\\', 'User', 'Documents', 'Mon_dossier', 'index.html')
```

Dire si le fichier existe ou non

In [68]:
```
p.exists()
```

Out[68]:
```
False
```

Dire s'il s'agit d'un dossier

In [72]:
```
p.is_dir()
```

Out[72]:
```
False
```

Dire s'il s'agit d'un fichier

In [73]:
```
p.is_file()
```

Out[73]:
```
False
```

Dossier parent

In [74]:
```
p.parent
```

Out[74]:
```
WindowsPath('/User/Documents/Mon_dossier')
```

Créer et supprimer des fichiers

```
```

In [76]:
```
# chemin
p = Path.home()

# Concaténation = nom du nouveau fichier
p = p / "DossierTest"

# création
p.mkdir()
```

Note : `mkdir()` ne fonctionnera pas si le dossier existe déjà, pour ne pas renvoyer d'erreur, il faut écrire :

`p.mkdir(exist_ok=True`

Il n'écrase pas le dossier existant, il n'affiche tout simplement pas l'erreur.

Créer toute une structure de dossiers, même si les dossiers parents n'existent pas :

In [81]:
```
p = p / "1" / "2" / "3"
```

In [83]:
```
p.mkdir(parents=True)
```

Créer un fichier

In [113…
```
# indique le chemin + nom du fichier
p = Path.home() / "Documents" / "readme.txt"

# création avec touch()
p.touch()
```

Supprimer un fichier

In [86]:
```
p.unlink()
```

Supprimer un dossier

Note : il faut que le dossier soit vide, pour effacer des dossiers non vides, il faut utiliser shutil

In [90]:
```
import shutil

shutil.rmtree(p)
```

In [102…
```
# pour supprimer un dossier vide :
p.rmdir()
```

Ecrire et lire dans un fichier

```
```

In [114…
```
# création du chemin
p = Path.home() / "Documents" / "readme.txt"
p
```

Out[114]:
```
WindowsPath('C:/Users/utilisateur/Documents/readme.txt')
```

In [116…
```
# création du fichier
p.touch()
```

In [117…
```
# écrire du texte
p.write_text("Bonjour")
```

Out[117]:
```
7
```

In [118…
```
# lire le contenu du fichier
p.read_text()
```

Out[118]:
```
'Bonjour'
```

Scanner un dossier

In [125…
```
from pathlib import Path

# Path.home().iterdir()
# for f in Path.home().iterdir():
#     print(f)

p = Path.home() / "Documents"

for fichier in p.iterdir():
    print(fichier)
```

```
C:\Users\utilisateur\Documents\.ipynb_checkpoints
C:\Users\utilisateur\Documents\AttestationDroits.pdf
C:\Users\utilisateur\Documents\biblio.odt
C:\Users\utilisateur\Documents\Coffres-forts McAfee
C:\Users\utilisateur\Documents\desktop.ini
C:\Users\utilisateur\Documents\GitHub
C:\Users\utilisateur\Documents\Ma musique
C:\Users\utilisateur\Documents\Mes images
C:\Users\utilisateur\Documents\Mes vidéos
C:\Users\utilisateur\Documents\microsoft_ia
C:\Users\utilisateur\Documents\model.h5
C:\Users\utilisateur\Documents\Modèles Office personnalisés
C:\Users\utilisateur\Documents\OpenCR_Cours.ipynb
C:\Users\utilisateur\Documents\projet_3_traitement_data.ipynb
C:\Users\utilisateur\Documents\Python Scripts
C:\Users\utilisateur\Documents\readme.txt
```

Possible de créer une liste

In [128…
```
# ne récupère que les dossiers
liste = [f for f in p.iterdir() if f.is_dir()]
print(liste)
```

```
[WindowsPath('C:/Users/utilisateur/Documents/.ipynb_checkpoints'), WindowsPath
('C:/Users/utilisateur/Documents/Coffres-forts McAfee'), WindowsPath('C:/Users/uti
lisateur/Documents/GitHub'), WindowsPath('C:/Users/utilisateur/Documents/Ma musiqu
e'), WindowsPath('C:/Users/utilisateur/Documents/Mes images'), WindowsPath('C:/Use
rs/utilisateur/Documents/Mes vidéos'), WindowsPath('C:/Users/utilisateur/Document
s/microsoft_ia'), WindowsPath('C:/Users/utilisateur/Documents/Modèles Office perso
nnalisés'), WindowsPath('C:/Users/utilisateur/Documents/Python Scripts')]
```

In [129…
```python
# ne récupère que les fichiers
liste = [f for f in p.iterdir() if f.is_file()]
print(liste)
```

```
[WindowsPath('C:/Users/utilisateur/Documents/AttestationDroits.pdf'), WindowsPath
('C:/Users/utilisateur/Documents/biblio.odt'), WindowsPath('C:/Users/utilisateur/D
ocuments/desktop.ini'), WindowsPath('C:/Users/utilisateur/Documents/model.h5'), Wi
ndowsPath('C:/Users/utilisateur/Documents/OpenCR_Cours.ipynb'), WindowsPath('C:/Us
ers/utilisateur/Documents/projet_3_traitement_data.ipynb'), WindowsPath('C:/Users/
utilisateur/Documents/readme.txt')]
```

Afficher que les fichiers en fonction de l'extension

In [135…
```python
for f in p.glob("*.pdf"):
    print(f.name)
```

```
AttestationDroits.pdf
```

In [145…
```python
# dossier téléchargement ou pictures

p = Path.home() / "Pictures"

for f in p.glob("*.PNG"):
    print(f.name)
```

```
By Ecodair.png
ré-inscription.PNG
```

Ajouter un suffixe à un nom de fichier

In [147…
```python
p = Path.home() / "Pictures" / "image.png"
p
```

Out[147]:
```
WindowsPath('C:/Users/utilisateur/Pictures/image.png')
```

In [150…
```python
p.parent / (p.stem + "-lowers" + p.suffix)
```

Out[150]:
```
WindowsPath('C:/Users/utilisateur/Pictures/image-lowers.png')
```

Trier des fichiers selon leur extension

In [157…
```python
from pathlib import Path

# dictionnaire avec extension et leur dossier de destination
dirs = {".png": "Images",
        ".jpeg": "Images",
        ".jpg": "Images",
```

```
        ".gif": "Images",
        ".mp4": "Videos",
        ".mp3": "Musiques",
        ".zip": "Archives",
        ".pdf": "Documents",
        ".txt": "Documents",
        ".json": "Documents"}

tri_dir = Path.home()/"Downloads"

# ne récupérer que les fichiers
files = [f for f in tri_dir.iterdir() if f.is_file()]

# déplacement des fichiers
for f in files :
    # si le fichier n'a pas d'extension dans le dico, met dans le dossier "autres"
    output_dir = tri_dir / dirs.get(f.suffix, "Autres")
    # création du fichier, s'il existe déjà, ne pas renvoyer d'erreur ni l'écraser
    output_dir.mkdir(exist_ok=True)
    # déplace le fichier = avec uniquement le nom du fichier (pas le chemin complet
    f.rename(output_dir /f.name)
```

In [ ]: