

Les fichiers

Se déplacer dans les répertoires

Lorsqu'un programme termine son exécution, toutes les informations stockées dans des variables sont perdues. Un moyen de les conserver est de les enregistrer dans un fichier sur un disque dur. A l'intérieur de celui-ci, les informations peuvent apparaître sous un format de texte (qui est alors lisible par divers logiciels et tout traitement de texte) mais on peut le stocker dans d'autres format (compressé ou en binaire ou codé).

Module os

Le module os permet quelques commandes du DOS (disk operating system) afin de pouvoir se déplacer dans l'arborescence de windows ou de linux. En particulier les fonctions `os.getcwd` et `os.chdir`.

```
import os as os
```

- `os.getcwd()` # Retourne le nom du répertoire courant
- `os.chdir("d:\\travail")` # Change le répertoire courant.

Remarque : le (backslash) est un caractère spécial, il doit être doublé dans le chemin d'accès du fichier ou remplacé par / (slash) : `os.chdir("d:/travail")`.

Manipuler les fichiers : écrire

Les fichiers texte sont les plus simples : ce sont des suites de caractères (format txt, mais aussi html, xml, csv...).

Ecriture

Les informations sont toujours écrites sous forme de chaînes de caractères et toujours ajoutées à la fin du fichier. L'écriture s'effectue toujours selon le même schéma

1. **création ou ouverture du fichier** → `f=open("nom-fichier";"w")` "w" :réécriture ou création
"a" : on complète le fichier
2. **écriture** → `f.write(s1)` écriture de la chaîne s1
`f.write(s2)` écriture de la chaîne s2 ...
3. **fermeture** → `f.close()`

Certains caractères sont très utiles pour organiser les données

- le symbole ; ou la tabulation "\t" sont souvent utilisés pour séparer des données
- "\n" indique un retour à la ligne.

On peut utiliser `print()` au lieu de `f.write()`, en gardant la syntaxe habituel. Seule modification, rajouter `file=f` pour que la sortie ait lieu dans le fichier et pas à l'écran.

```
print(v1, v2, ..., sep=' ', end='\n', file=f)
```

Remarque : avec "w", le fichier est réécrit complètement, avec "a", on le complète.

Manipuler les fichiers : lire

Lorsque l'on veut utiliser un fichier le plus simple est de le transférer en mémoire, de le manipuler, puis éventuellement de réécrire le fichier modifié.

Pour cela, on crée une liste T dont chaque élément est une ligne du fichier (la lecture du fichier se fait ligne par ligne).

```
f=open("nom-fichier";"r") # ouverture du fichier en mode lecture
T=[] # T contiendra le fichier en mémoire
for ligne in f : # pour toutes les lignes du fichier
    T.append(ligne) # on peut en profiter pour nettoyer les caractères parasites!
f.close() # on ferme le fichier
```

Remarque : On peut stocker le fichier en mémoire (variable T) avec une seule instruction : `T=f.readlines()`

Remarque : une fois le fichier en mémoire, on se débarrasse des caractères parasites "\t", "\n", "\r" ou ";" (pensez aux fonctions `s.remove()` et `s.split()`).