

# Leçon d'informatique : Fichiers

S. Benhajlahsen - PCSI<sub>1</sub>



## Sommaire

I Fichiers et répertoires informatique	1
II Précautions d'utilisation	2
III Lecture d'un fichier	2
IV Écriture dans un fichier	4
V Passer un fichier en argument d'entrée	4

**Introduction** Le traitement de données numériques passe souvent par la manipulation de fichiers : en lecture quand on veut récupérer des données déjà écrites dans un fichier, ou en écriture quand on veut produire un tel fichier. En outre, il est extrêmement courant de placer la configuration d'un logiciel dans un fichier, auquel il faut pouvoir accéder en lecture ou en écriture.

## I Fichiers et répertoires informatique

**Fichiers informatiques** Un fichier<sup>a</sup> est un ensemble de données numériques réunies sous un même nom.

*a. file en anglais*

**Dossier ou répertoire** En vue de faciliter leur organisation, les fichiers sont disposés dans des **systèmes de fichiers** (voir figure 1) qui permettent de placer les fichiers dans des emplacements appelés **répertoires** ou **dossiers** eux-mêmes organisés selon le même principe de manière à former une hiérarchie arborescente.

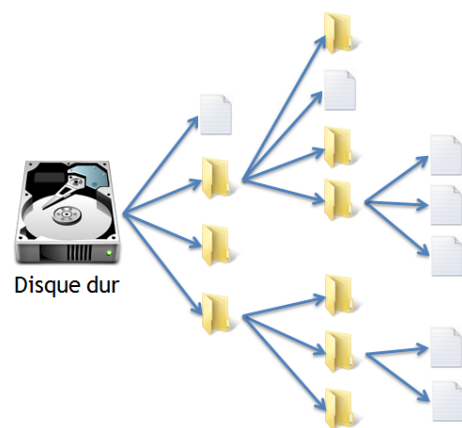
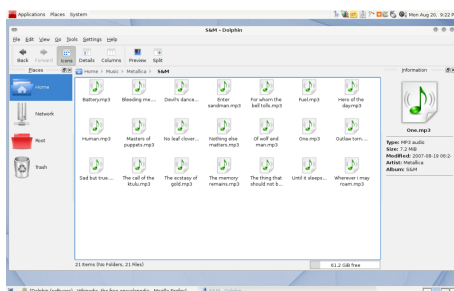


FIGURE 1 – À gauche, système de fichiers. À droite, arborescence de fichiers.

**Format de fichier** Le format de fichier est la convention selon laquelle les informations sont numérisées et organisées dans le fichier. Généralement, le format est indiqué par l'**extension** qui est suffixe sur le nom. On retiendra donc que les fichiers sont écrits sous la forme :

`nom.extension`

Par exemple :

- `tp1.py` est un fichier exécutable par Python et dont le nom est `tp1`.
- `nom.txt` est un simple fichier texte non formaté.
- `nom.docx` est un fichier lisible par le logiciel propriétaire Word de Microsoft.
- `nom.pdf` est un fichier pdf (*portable document format*) qui est un format assez universel, lisible et imprimable facilement.

**Chemin d'accès** L'organisation des fichiers<sup>a</sup> d'un fichier ou d'un répertoire est une chaîne de caractères décrivant la position de ce fichier ou répertoire dans le système de fichiers. Pour :

- les systèmes d'exploitation Linux/MacOS, celui-ci part du **répertoire racine** / selon la forme :

```
/Users/Documents/lecons/10_fichiers.pdf
```

Cela veut que, partant de la racine<sup>b</sup>, on choisit le répertoire `Users/` qui contient le sous-répertoire `Documents` puis `lecons/` qui contient le fichier `10_fichiers.pdf`.

- les systèmes Windows, celui-ci commence par une lettre pour indiquer le lecteur :

```
C:\Users\Documents\lecons\10_fichiers.pdf
```

a. *path* en anglais  
b. *root* en anglais

## II Précautions d'utilisation

### II.A Ouverture

On ne peut pas lire ou écrire dans un fichier sans précaution<sup>1</sup> :

- Demande d'ouverture : Python vérifie que vous avez les **permissions**<sup>2</sup> nécessaires dans le dossier où il se trouve, et demande au système d'exploitation l'accès au fichier.
- Demande de fermeture : les dernières opérations sur le fichier sont effectuées, puis la variable représentant le fichier est détruite.

**Syntaxe** : La syntaxe d'ouverture d'un fichier est :

```
cur = open(chemin, mode)
```

`cur` est la variable qui représentera le **curseur** dans le fichier dans la suite du programme.

`chemin` est la chaîne de caractère contenant le chemin du fichier.

`mode` est une chaîne de caractère : `"r"` pour ouvrir en lecture (*read*), `"w"` pour ouvrir en écriture (*write*).

### II.B Fermeture

**Syntaxe** : La syntaxe de fermeture d'un fichier représenté par la variable fichier est :

```
cur.close()
```

## III Lecture d'un fichier

**Exemple** Supposons que le fichier `nom_premiers.txt` soit dans le dossier `"/Users/soufianebenhajlahsen/Documents/"`. Si on l'ouvrait dans un éditeur de texte<sup>a</sup>, on verrait la figure 2.

a. Comme TextEdit sous macOS, Notepad sous Windows ou Gedit sous Linux.

**Etape primordiale** Il faut indiquer à Python le dossier de travail<sup>a</sup>. Ce dossier peut être différent du dossier dans lequel le fichier est enregistré. Vous pouvez vérifier quel est le `working directory` dans le navigateur de fichiers<sup>b</sup> de Pyzo (voir figure 3).

1. En outre, ce chapitre se limite aux « fichiers texte », donc aux fichiers que vous pourriez aussi ouvrir avec un éditeur de texte. Leur contenu doit pouvoir se ramener à une (ou plusieurs) chaînes de caractères.

2. ou droits d'accès aux ressources

On peut régler le dossier de travail pour tout le programme avec l'instruction de la figure 4.

- a. *working directory*
- b. *file browser* en anglais

2

3  
5  
7  
11

FIGURE 2 – nombres premiers

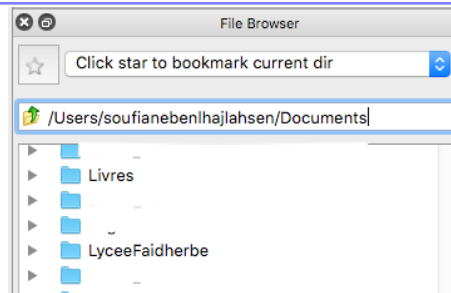


FIGURE 3

```
1 import os
2 os.chdir("/Users/soufianebenhajlahsen/Documents")
```

FIGURE 4

**Lecture en masse** La syntaxe de la figure 5 qui lit tout le fichier et le renvoie sous forme d'une chaîne de caractères. Le caractère `\n` indique la fin de la ligne. Soit ici :

```
contenu = "2\n3\n5\n7\n11"
```

**Lecture ligne par ligne, d'un seul coup** La syntaxe de la figure 6 qui renvoie la liste des lignes. Soit ici :

```
contenu = ['2\n', '3\n', '5\n', '7\n', '11\n']
```

**Boucler directement sur le fichier** La variable représentant le curseur peut servir de liste dans une boucle `for`. Dans l'exemple précédent, la sortie de `readlines` n'est pas vraiment pratique : on aimerait éliminer les `"\n"`. et convertir les chaînes de caractères restantes en nombres entiers. On pourrait bien sûr le faire après-coup en bouclant sur la liste obtenue ou exécuter le programme de la figure 7. On obtient alors :

```
[2, 3, 5, 7, 11]
```

#### IV Écriture dans un fichier

Un fichier ouvert en **mode écriture** avec "w" est initialement vide<sup>3</sup>. Ensuite, `write` permet d'y ajouter des chaînes de caractères. Par exemple, pour créer le fichier `nombres_premiers.txt` utilisé précédemment, on a exécuté le programme de la figure 8.

3. Si le fichier existait déjà, son contenu est détruit!!!

---

```
1 import os
2 os.chdir("/Users/soufianebenlhajlahsen/Documents")
3 cur = open("nombres_premiers.txt", "r")
4 contenu = cur.read()
5 print(contenu)
6 cur.close()
```

---

FIGURE 5

---

```
1 import os
2 os.chdir("/Users/soufianebenlhajlahsen/Documents")
3 cur = open("nombres_premiers.txt", "r")
4 contenu = cur.readlines()
5 print(contenu)
6 cur.close()
```

---

FIGURE 6

## V Passer un fichier en argument d'entrée

La variable représentant le curseur fonctionne comme une liste si on la passe en argument d'entrée à une fonction : il n'y a pas création d'une variable locale, donc la fonction modifie directement le fichier, les modifications n'étant pas annulées quand la fonction se termine (voir figure 9).

---

```
1 import os
2 os.chdir("/Users/soufianebenlhajlahsen/Documents")
3 cur = open("nombres_premiers.txt", "r")
4 contenu = []
5
6 for lignes in cur:
7     res = int(lignes[0:-1])
8     contenu.append(res)
9 print(contenu)
10 cur.close()
```

---

FIGURE 7

---

```
1 import os
2 os.chdir("/Users/soufianebenlhajlahsen/Documents")
3 L = [2,3,5,7,11]
4
5 cur = open("nombres_premiers.txt", "w")
6 for n in L :
7     cur.write(str(n)+"\n")
8 cur.close()
```

---

FIGURE 8

---

```
1 import os
2 os.chdir("/Users/soufianebenlhajlahsen/Documents")
3
4 # ecrit le nombre n avec saut de ligne
5 # dans le fichier fich
6
7 def ecrire(fich, n):
8     fich.write(str(n)+"\n")
9
10
11 cur = open("nombres_premiers.txt", "w")
12 L = [2,3,5,7,11]
13 for n in L:
14     ecrire(cur, n)
15 cur.close()
```

---

FIGURE 9