

# LES FICHIERS

**OBJECTIFS**

- savoir définir un fichier informatique
- savoir lire et écrire dans un fichier
- savoir charger en mémoire le contenu d'un fichier CSV
- réviser les listes, les tuples et le formatage de chaînes de caractères

## ➤ Connaissances et savoir-faire à maîtriser

### 1) Définition

Un fichier est une donnée numérique enregistrée sur un support de stockage non volatile, c'est-à-dire une mémoire qui survit à une absence d'alimentation électrique comme un disque dur ou une clé USB par exemple. On distingue deux types de fichiers en informatique :

- **les fichiers textes** qui contiennent du texte "compréhensible" par un être humain (extension en .txt, .xml, .csv, ...);
- **les fichiers binaires** qui contiennent des suites d'octets et qui ne sont compréhensible que par un programme spécifique qui "sait" comment les interpréter (extension en .exe, .doc, .pdf, ...).

*Remarque 1 : l'extension ne définit pas le fichier en lui-même mais ne sert qu'à indiquer au système d'exploitation le logiciel avec lequel il doit ouvrir le fichier. Vous pouvez vous en rendre compte en changeant l'extension d'un fichier PDF en .py ou .html.*

*Remarque 2 : étant donné qu'en informatique tout n'est qu'une succession de 0 et de 1, un fichier texte n'est en réalité qu'une forme particulière de fichier binaire.*

### 2) Manipulation de fichiers

Pour échanger des données entre votre programme et un fichier, il est nécessaire d'ouvrir un **flux d'entrée/sortie** sur ce fichier. Python offre un moyen pratique de gérer les flux E/S vers les fichiers :

```
with open("data.txt", 'r') as fic: # ouverture du fichier en lecture
    ... # opérations sur fic
```

*Remarque 3 : le paramètre 'r' doit être remplacé par 'w' pour une ouverture en écriture.*

*Remarque 4 : on se limitera dans ce TP à la manipulation des fichiers textes.*

#### a. Lire dans un fichier

```
with open("data.txt", 'r') as fic:
    contenu = fic.read().splitlines() # contenu est de type list
```

Une fois le contenu du fichier chargée dans la variable contenu, vous pourrez opérer sur cette variable qui contient une liste dont les éléments sont des chaînes de caractères représentant chaque ligne du fichier.

## b. Écrire dans un fichier

```
with open("data.txt", 'w') as fic:
    fic.write("Hello World !!!\n") # \n saute une ligne
```

On écrit dans un fichier pour garder une trace des données du programme une fois celui-ci terminé (on parle de **persistance des données**). Pour notre TP de mini-système d'authentification, ce serait une manière simple et pratique de garder une trace du dictionnaire entre chaque exécution du programme.

## c. Les fichiers CSV

Un fichier CSV pour *Comma-separated values* est un fichier texte stockant des données sous forme de valeurs séparées par des virgule. Le contenu de notre dictionnaire de mot de passe se présenterai par exemple comme ceci :

```
alice,ab4f63f9ac65152575886860dde480a1
bob,827ccb0eea8a706c4c34a16891f84e7b
carol,b6edd10559b20cb0a3ddaeb15e5267cc
```

Une fois le fichier lu ligne par ligne, un découpage de chaque chaîne avec la méthode de chaîne `str.split()` permet d'en extraire les informations dans le but de les stocker dans une variable de votre programme.

*Remarque 5 : en pratique, vous pouvez choisir n'importe quel caractère comme séparateur, il suffit simplement que ce caractère ne soit pas présent dans vos valeurs, tout comme vous pouvez donner l'extension qui vous plaît à votre fichier voire même pas d'extension du tout.*

## 🔗 Exercices à réaliser

### EXERCICE 1

Ouvrir le fichier TP-fichiers-exo1.py et y copier-coller le code ou son corrigé du TP sur le mini-système d'authentification. Copier ensuite le fichier `comptes.txt` **dans le même dossier** que TP-fichiers-exo1.py puis répondre aux questions suivantes :

1. Écrire une fonction `lire_dico(pnomfic)` qui crée un dictionnaire vide et le remplit du contenu du fichier `comptes.txt` correctement stocké sous la forme clé/valeur.

```
lire_dico(pnomfic)
|- ENTREES : pnomfic est de type str
|- SORTIES : un dictionnaire de couple user/password
```

Lire ce fichier à chaque démarrage de l'application pour charger le dictionnaire des comptes.

2. Écrire une fonction `enregister_dico(pnomfic, pdico)` qui stocke les couples clé/valeur de `pdico` dans le fichier `pnomfic`.

```
enregister_dico(pnomfic, pdico)
|- ENTREES : pnomfic est de type str, pdico est de type dictionnaire
|- SORTIES : rien
```

Faire en sorte d'enregistrer ce dictionnaire lorsque l'utilisateur quitte le programme afin d'assurer la persistance des données.

**EXERCICE 2**

L'objet de cet exercice est d'exploiter un fichier de données obtenu à partir du site de l'INSEE. Copier le fichier `data.csv` dans votre répertoire de travail puis ouvrir le fichier `TP-fichiers-exo2.py`, lire les consignes et répondre aux questions suivantes :

- Après avoir observé la structure et la nature des données contenues dans le fichier `data.csv`, écrire une fonction `afficher_fichier(pnomfic)` qui extrait, formate et affiche ces données :

```
afficher_fichier(pnomfic)
    ENTREES : pnomfic est de type str
    SORTIES : rien
```

- Écrire une fonction `charger_fichier(pnomfic, pliste)` qui charge le contenu du fichier nommé `pnomfic` dans la variable `pliste` et où chaque élément de la liste est un tuple de la forme (DEP, LIBDEP, TN19, AGE19, NAISS18).

```
charger_fichier(pnomfic, pliste)
    ENTREES : pnomfic est de type str, pliste est une liste de tuples
    SORTIES : rien
```





*Attention, les valeurs de TN19, AGE19 et NAISS18 devront être stockées dans le tuple sous la forme de flottants ou d'entiers.*

- Écrire une fonction `calculer_total_naissance(pliste)` qui calcule et renvoie le nombre total de naissances en France en 2018 :

```
calculer_total_naissance(pliste)
    ENTREES : pliste est une liste de tuples
    SORTIES : un entier représentant le nombre total de naissances en France 2018
```

- Écrire deux fonctions permettant respectivement de déterminer le département ayant le plus fort et le plus faible taux de natalité en 2019.

## 🔗 Compétences à valider

LES FICHIERS					
THÉORIE	Je sais définir un fichier binaire				
	Je sais définir un fichier texte				
	J'ai compris le rôle d'une extension de fichier				
	Je connais la structure interne d'un fichier CSV				
PRATIQUE	Je sais écrire dans un fichier				
	Je sais lire et afficher le contenu d'un fichier CSV				
	Je sais charger le contenu d'un fichier CSV en mémoire afin d'en manipuler les données				
	Je maîtrise les listes, les tuples et le formatage de chaînes de caractères				