



# Table des matières

Table des matières..... 1

1. Les fichiers ..... 1

2. Exploitation des fichiers..... 3

## 1. Les fichiers

### 1.1. Définition

Un **fichier informatique** est au sens commun, une collection, un ensemble de données **numériques** réunies sous un même nom, enregistrées sur un support de stockage permanent : carte mémoire, disque dur, ssd .....

Un fichier est caractérisé par son nom et son suffixe (ou extension) qui caractérise sur la nature des informations contenues :

### 1.2. Les principaux formats de fichiers





## 1.3. Quelques formats ouverts de données

Un format de données est dit ouvert (ou libre) si son mode d'organisation a été rendu public par son auteur et qu'aucune entrave légale ne s'oppose à sa libre utilisation (droit d'auteur, brevet, copyright).



### **.XML ou Extensible Markup Language**

Format utilisé pour l'échange automatisé entre système d'information



### **.CSV : Comma Separated Values**

Format de texte où chaque valeur est généralement séparée par une virgule, ce qui permet d'enregistrer les données dans un tableau



### **.VCF : Virtual Card File**

Format de carnet d'adresse qui peut contenir une ou plusieurs adresses

## 1.4. Emplacement

### 1.4.1. Chemin absolu

Le chemin absolu d'un fichier (appelé path en anglais) est l'endroit exact où l'on peut trouver le fichier sur la machine. Par exemple un fichier «Projet\_1.pdf» qui serait placé dans un dossier appelé « Projets » lui-même placé dans MesDocuments du PC a un chemin absolu :

```
C:\Users\NomUtilisateur\Documents\Projets \Projet_1.pdf
```

C: désignant la racine du support de stockage ( sous Windows C: est le support de stockage où se trouve Windows )

Pour l'appel à un fichier dans python ou tout autre langage utilisant des fichiers extérieurs (HTML et CSS par exemple), on peut utiliser le chemin absolu avec la certitude de ne pas faire d'erreur... sauf si on déplace notre dossier de travail, sur une clé, sur un autre pc ...

### 1.4.2. Chemin relatif

Le chemin relatif donne le chemin du fichier que l'on cherche par rapport à celui dans lequel on travaille. Si le fichier .py est dans le même répertoire (dossier) que le fichier Projet\_1.pdf, on peut se contenter d'écrire «Projet\_1.pdf» pour ouvrir le fichier.

. désigne le répertoire lui-même

.. désigne le répertoire parent



## 2. Exploitation des fichiers

Soit le fichier Commune.csv :

```
Code_commune_INSEE;Nom_commune;Code_postal;Libelle_acheminement;Ligne_5;coordonnees_gps
90093;SERMAMAGNY;90300;SERMAMAGNY;;47.687801557,6.8309146345
91093;BOULLAY LES TROUX;91470;BOULLAY LES TROUX;;48.6753515056,2.04828313772
91100;BOUVILLE;91880;BOUVILLE;;48.4326483441,2.2783856422
91129;CERNY;91590;CERNY;;48.4859798517,2.31068283872
```

### 2.1. Ouverture d'un fichier

#### 2.1.1. La fonction open

##### 2.1.1.1. Ouverture du fichier



Les paramètres de la fonction `open("Nom du Fichier", 'attribut')` les différents attribus sont :

r	Ouvre un fichier en lecture uniquement. Le pointeur est placé au début du fichier. C'est la valeur par défaut.
w	Ouvre un fichier en écriture uniquement. Écrase le contenu du fichier s'il existe déjà, crée un nouveau fichier sinon. Le pointeur est placé au début du fichier.
a	Ouvre un fichier en écriture uniquement. Le pointeur est placé à la fin du fichier s'il existe, sinon un nouveau fichier est créé.
rb ou wb ou ab	Ouvre le fichier en mode binaire (lecture des octets au lieu des caractères).
r+ ou w+ ou a+	Ouvre le fichier en lecture et en écriture.
rb+ ou wb+ ou ab+	Combinaison des deux précédents.
x	Crée un nouveau fichier et l'ouvre pour écriture



### 2.1.1.2. Exploitation du contenu

- Lecture de tout le fichier méthode `str.read()`

- Lecture d'une ligne méthode `str.readline()`

Sortie :

Code\_commune\_INSEE;Nom\_commune;Code\_postal;Libelle\_acheminement;Ligne\_5;coordonnees\_gps

90093;SERMAMAGNY;90300;SERMAMAGNY;;47.687801557,6.8309146345

- Lecture du reste des lignes méthode `str.readlines()`

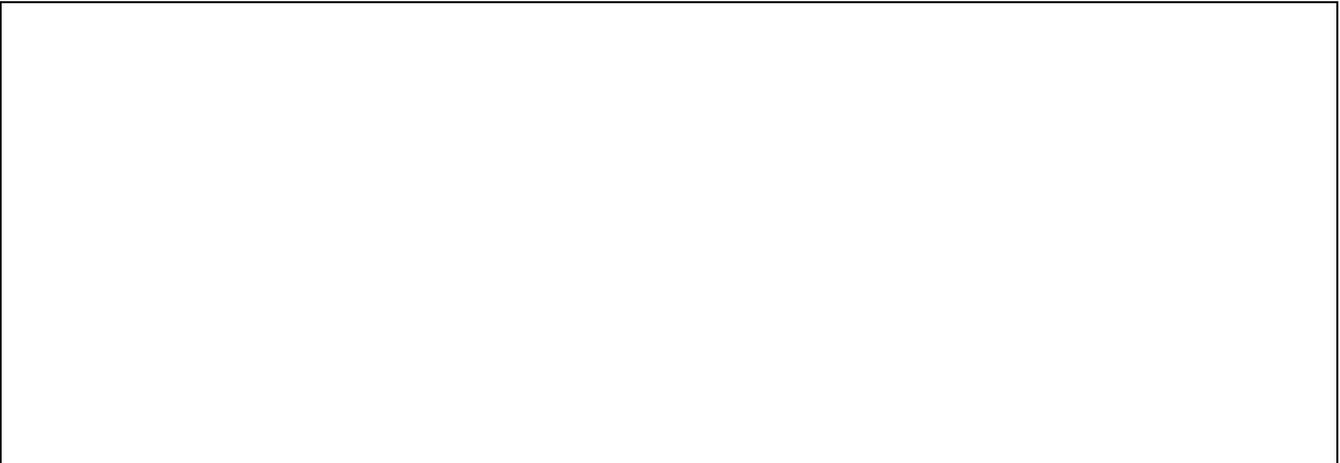
Sortie : Code\_commune\_INSEE;Nom\_commune;Code\_postal;Libelle\_acheminement;Ligne\_5;coordonnees\_gps

90093;SERMAMAGNY;90300;SERMAMAGNY;;47.687801557,6.8309146345

['91093;BOULLAY LES TROUX;91470;BOULLAY LES TROUX;;48.6753515056,2.04828313772\n',  
'91100;BOUVILLE;91880;BOUVILLE;;48.4326483441,2.2783856422\n', '91129;CERNY;91590;CERNY;;48.4859798517,2.31068283872\n']



- Parcours du fichier ligne par ligne jusqu'à atteindre une ligne vide



Sortie : Code\_commune\_INSEE;Nom\_commune;Code\_postal;Libelle\_acheminement;Ligne\_5;coordonnees\_gps

90093;SERMAMAGNY;90300;SERMAMAGNY;;47.687801557,6.8309146345

91093;BOULLAY LES TROUX;91470;BOULLAY LES TROUX;;48.6753515056,2.04828313772

91100;BOUVILLE;91880;BOUVILLE;;48.4326483441,2.2783856422

91129;CERNY;91590;CERNY;;48.4859798517,2.31068283872

## 2.1.2. Le mot clé with

- Ouverture



Sortie :

Code\_commune\_INSEE;Nom\_commune;Code\_postal;Libelle\_acheminement;Ligne\_5;coordonnees\_gps

90093;SERMAMAGNY;90300;SERMAMAGNY;;47.687801557,6.8309146345

91093;BOULLAY LES TROUX;91470;BOULLAY LES TROUX;;48.6753515056,2.04828313772

91100;BOUVILLE;91880;BOUVILLE;;48.4326483441,2.2783856422

91129;CERNY;91590;CERNY;;48.4859798517,2.31068283872

- Parcours ligne par ligne



Sortie :

Code\_commune\_INSEE;Nom\_commune;Code\_postal;Libelle\_acheminement;Ligne\_5;coordonnees\_gps

90093;SERMAMAGNY;90300;SERMAMAGNY;;47.687801557,6.8309146345

91093;BOULLAY LES TROUX;91470;BOULLAY LES TROUX;;48.6753515056,2.04828313772

91100;BOUVILLE;91880;BOUVILLE;;48.4326483441,2.2783856422



```
91129;CERNY;91590;CERNY;;48.4859798517,2.31068283872
```

## 2.2. Ecriture dans un fichier

Ecrit une chaîne de caractère dans le fichier et retourne le nombre de caractères écrits

Sortie : Nombre de caractères écrits 9

## 2.3. Cas particulier les fichiers csv

Les fichiers **Comma Separated Values .csv** sont des fichiers dont les différents champs d'une ligne sont séparés par une virgule, un point-virgule ou une tabulation.

Pour exploiter leurs données on utilise la méthode `str.split(separator, maxsplit)` qui permet de :

- Séparer les éléments de la chaîne de caractères par rapport au séparateur
- De retourner une liste des éléments de la chaîne de caractères séparés

**separator** : est le délimiteur. La chaîne se divise à ce séparateur spécifié. Si n'est pas fourni, tout espace blanc est un séparateur.

**maxsplit** : est un nombre qui nous dit de scinder la chaîne en un nombre maximum de fois. S'il n'est pas fourni, il n'y a pas de limite.

Sortie :

```
['Code_commune_INSEE', 'Nom_commune', 'Code_postal', 'Libelle_acheminement', 'Ligne_5', 'coordonnees_gps\n']  
['90093', 'SERMAMAGNY', '90300', 'SERMAMAGNY', '', '47.687801557,6.8309146345\n']  
['91093', 'BOULLAY LES TROUX', '91470', 'BOULLAY LES TROUX', '', '48.6753515056,2.04828313772\n']  
['91100', 'BOUVILLE', '91880', 'BOUVILLE', '', '48.4326483441,2.2783856422\n']  
['91129', 'CERNY', '91590', 'CERNY', '', '48.4859798517,2.31068283872\n']
```



Il existe aussi la bibliothèque [Lib/csv.py](#) qui permet de manipuler les fichiers au format csv avec des outils adaptés.

```
import csv
spamreader = open("eggs.csv", 'w+')

with open('eggs.csv', 'w', newline='') as csvfile:
    spamwriter = csv.writer(csvfile, delimiter=' ',
                            quotechar='|', quoting=csv.QUOTE_MINIMAL)
    spamwriter.writerow(['Spam'] * 5 + ['Baked Beans'])
    spamwriter.writerow(['Spam', 'Lovely Spam', 'Wonderful Spam'])

with open('eggs.csv', newline='') as csvfile:
    spamreader = csv.reader(csvfile, delimiter=' ', quotechar='|')
    for row in spamreader:
        print(', '.join(row))
```

Sortie :

Spam, Spam, Spam, Spam, Spam, Baked Beans

Spam, Lovely Spam, Wonderful Spam

Dans le fichier edds.csv

```
Spam Spam Spam Spam Spam |Baked Beans|
Spam |Lovely Spam| |Wonderful Spam|
```