Lire et écrire dans un fichier texte

Objectifs :

- Savoir **ouvrir, lire et écrire** dans un fichier texte.
- Comprendre la différence entre les **modes** `"r"` (lecture), `"w"` (écriture), `"a"` (ajout).
- Traiter des fichiers **ligne par ligne**.
- Savoir **fermer** un fichier proprement (ou utiliser l'instruction `with open(…)`).

Table des matières

I. Lire et écrire dans un fichier texte en Python....1
1. Pourquoi utiliser des fichiers ?....1
2. Ouvrir un fichier : les différents modes....1
3. Lire le contenu d'un fichier ligne par ligne....1
a. Méthode 1 : lecture simple....2
b. Méthode 2 : lecture avec readlines()....2
4. Écrire dans un fichier....2
a. Écrasement du fichier si "w"....2
5. Ajouter du contenu avec "a"....2
6. Utilisation recommandée : with....2
7. Quelques fonctions utiles....2
8. À retenir....2

I. Lire et écrire dans un fichier texte en Python

1. Pourquoi utiliser des fichiers ?

Un fichier texte permet de :

- **sauvegarder** des données (par exemple, les résultats d'un programme) ;
- **recharger** des données plus tard ;
- **échanger** des informations entre programmes ou avec l'utilisateur.

2. Ouvrir un fichier : les différents modes

```
open(nom_fichier, mode)
```

Mode	Signification
`"r"`	lecture (read)
`"w"`	écriture (write, efface si existe)
`"a"`	ajout (append à la fin du fichier)

3. Lire le contenu d'un fichier ligne par ligne

Supposons un fichier `texte.txt` avec seulement trois lignes de contenu :

```
Bonjour
Python
Fichier
```

a. Méthode 1 : lecture simple

```
f = open("texte.txt", "r")
for ligne in f:
    print(ligne.strip())  # .strip() supprime le \n
f.close()
```

b. Méthode 2 : lecture avec `readlines()`

```
f = open("texte.txt", "r")
lignes = f.readlines()
print(lignes)
f.close()
```

4. Écrire dans un fichier

a. Écrasement du fichier si "w"

```
f = open("resultat.txt", "w")
f.write("Première ligne\n")
f.write("Deuxième ligne\n")
f.close()
```

5. Ajouter du contenu avec "a"

```
f = open("resultat.txt", "a")
f.write("Nouvelle ligne ajoutée\n")
f.close()
```

6. Utilisation recommandée : `with`

```
with open("texte.txt", "r") as f:
    for ligne in f:
        print(ligne.strip())
```

Avantage : le fichier est automatiquement **fermé**.

7. Quelques fonctions utiles

Fonction/méthode	Description
`.read()`	lit tout le fichier
`.readline()`	lit une seule ligne
`.readlines()`	lit toutes les lignes → liste de str
`.write(texte)`	écrit une chaîne dans le fichier
`open(..., mode)`	ouvre un fichier dans un certain mode
`with open(...) as`	ouvre un fichier de façon sécurisée

8. À retenir

- Toujours fermer un fichier après usage (`f.close()`), sauf si on utilise `with`.
- `w` **efface** le contenu existant. `a` **ajoute** à la fin.
- `readlines()` permet d'avoir une **liste** des lignes du fichier.