



BAC Pro CIEL	<h2>Python</h2> <h3>Manipulation de fichiers</h3>	 Année 2025/2026
		

A. Pourquoi manipuler des fichiers ?

En Python, travailler avec des fichiers permet de :

- ⇒ Sauvegarder des données (logs, résultats, ...)
- ⇒ Lire des données (config, trames, historiques, ...)
- ⇒ Faire du traitement réel (cyber, réseau, data, ...)

B. Ouvrir un fichier : open()

```
fichier = open("mon_fichier.txt", "r")
```

Modes principaux :

"r"	Lecture (read)
"w"	Écriture, écrase le fichier existant
"a"	Ajout à la fin du fichier
"r+"	Lecture + écriture

C. Fermer un fichier

```
fichier = open("mon_fichier.txt", "r")
```

```
fichier.close()
```

Bonnes pratiques à privilégier : with

```
with open("mon_fichier.txt", "r") as fichier :
    contenu = fichier.read()
```

Avantages :

- ⇒ Fermeture automatique
- ⇒ Code plus propre

D. Lire un fichier

Lire tout le contenu du fichier :

```
with open("mon_fichier.txt", "r") as f :  
    contenu = f.read()  
    print(contenu)
```

Lire ligne par ligne :

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

.strip() enlève les retours à la ligne

Lire dans une liste

```
with open("mon_fichier.txt", "r") as f:  
    lignes = f.readlines()  
print(lignes)
```



E. Ecrire dans un fichier

Écriture simple (Attention écrase le fichier)

```
with open("mon_fichier.txt", "w") as f:  
    f.write("Bonjour\n")
```

Ajout du contenu

```
with open("mon_fichier.txt", "a") as f:  
    f.write("Nouvelle ligne\n")
```

Exemple concret :

Sauvegarder des données de capteurs :

```
with open("donnees.txt", "a") as f:  
    temperature = 23  
    f.write(f"Température : {temperature}°C\n")
```

F. Vérifier si un fichier existe

```
import os  
  
if os.path.exists("mon_fichier.txt"):  
    print("Le fichier existe")  
else:  
    print("Fichier introuvable")
```

G. Erreurs fréquentes

Oublier de sauvegarder le script dans le même répertoire que le fichier :

```
open("fichier.txt", "r")
```

Oublier le mode :

```
open("fichier.txt")
```

Ecraser sans le vouloir :

```
open("fichier.txt", "w")
```

Dès le lancement de cette instruction, le fichier « fichier.txt » est détruit (remis à zéro).

Fichier introuvable :

```
with open("fichier_inexistant.txt", "r") as f:  
    contenu = f.read()
```

FileNotFoundError

H. A retenir

`open()` permet d'ouvrir un fichier

"r" lit un fichier

"w" écrit mais efface l'ancien contenu

"a" ajoute les données à la fin du fichier

`with` ferme automatiquement le fichier

I. Synthèse

