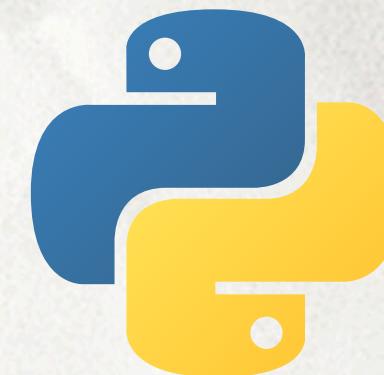


# Tutoriel 5

## Manipulation des fichiers avec Python



# **Pourquoi travailler avec des fichiers ?**

Travailler avec des fichiers est une compétence essentielle en programmation. Cela permet de gérer des données de manière durable, même après l'arrêt d'un programme. Les applications pratiques de la manipulation des fichiers sont nombreuses et très variées :

## **1. Stockage de données utilisateur :**

- Par exemple, conserver les préférences ou les scores d'un utilisateur dans un jeu ou une application.

## **2. Creation de journaux (logs) :**

- Les fichiers peuvent enregistrer l'historique d'événements (comme les connexions des utilisateurs ou les erreurs dans une application). Ces journaux aident les developpeurs a comprendre et resoudre les problèmes.

## **3. Gestion de bases de donnees legeres :**

- Avant d'utiliser une base de donnees complexe, des fichiers simples (comme un fichier texte ou JSON) peuvent stocker des informations structurees.

## **4. Partage et exportation de données :**

- Les fichiers permettent de transmettre des informations entre différents programmes ou personnes, comme des rapports ou des données analytiques au format CSV.

## **5. Crédit d'automatismes :**

- Gérer des fichiers pour automatiser des tâches répétitives : renommage de fichiers en masse, extraction de données dans un dossier, etc.

# Ouvrir et lire un fichier

Syntaxe basique pour lire un fichier texte.

```
with open('example.txt', 'r') as file:  
    contenu = file.read()  
    print(contenu)
```

# Écrire dans un fichier

Créer ou modifier un fichier texte.

```
with open('example.txt', 'w') as file:  
    file.write("Bonjour, Python !")
```

## **Erreurs courantes :**

- Fichiers manquants, permissions. Inclure des astuces pour les résoudre.

## **Exercice pour s'entraîner :**

- Créez un fichier texte et écrivez-y une liste de tâches simples.