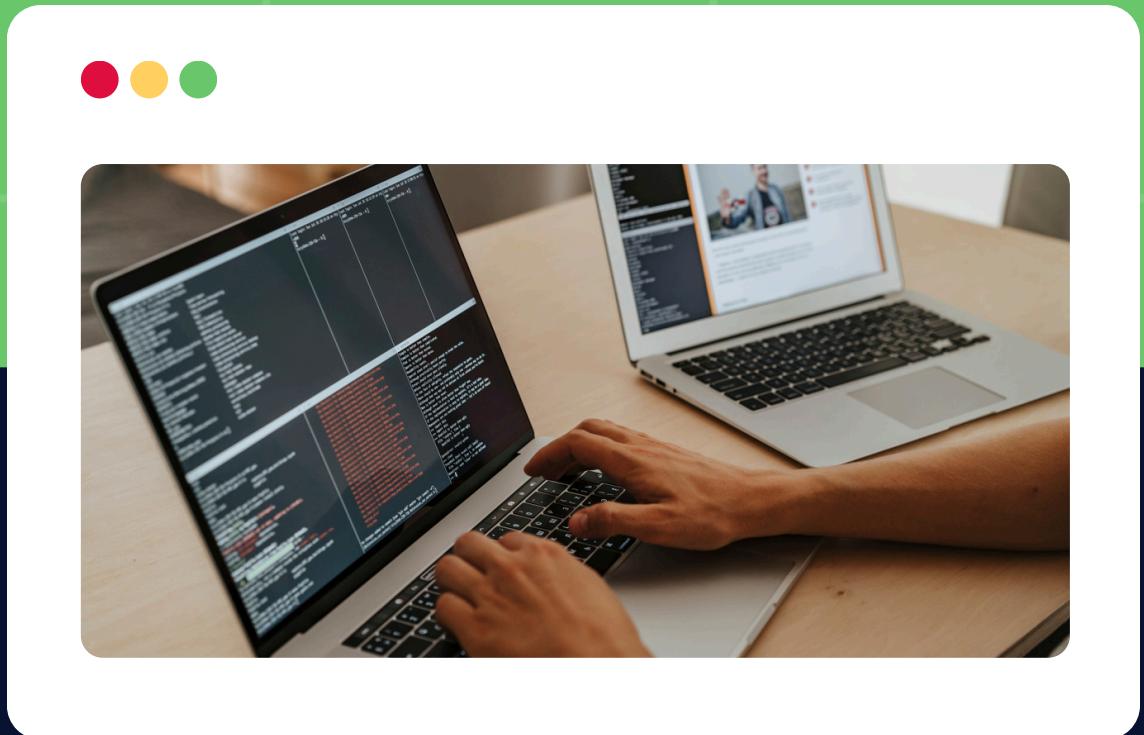




Lecture et écriture de fichiers



```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title= "product" />
          <div className="row">
            <ProductConsumer>
              {(value) => {
                console.log(value)
              }}
            </ProductConsumer>
          </div>
        </div>
      <React.Fragment>
    
```

Laffet takwa

Pourquoi manipuler des fichiers ?



Les fichiers servent à :

- Sauvegarder des données (ex : notes, utilisateurs, logs)
- Lire des informations depuis un fichier (ex : configurations)
- Échanger des données entre programmes (ex : CSV ou JSON)

Lecture et écriture dans un fichier texte (.txt)

Ouvrir un fichier : la fonction open()

Mode	Signification
"r"	Lecture (read)
"w"	Écriture (write) → écrase le contenu
"a"	Ajout (append) → ajoute à la fin
"x"	Crée un nouveau fichier (erreur si existe déjà)

Écrire dans un fichier texte

```
# Ouverture du fichier en mode écriture
f = open("exemple.txt", "w")

# Écriture de contenu
f.write("Bonjour\n")
f.write("Bienvenue dans le cours Python !")

# Fermeture du fichier
f.close()
```

Remarque : si le fichier existe déjà, son contenu sera écrasé.

Lire le contenu d'un fichier texte

a) Lire tout le contenu :

```
f = open("exemple.txt", "r")
contenu = f.read()
print(contenu)
f.close()
```

Lire le contenu d'un fichier texte

b) Lire ligne par ligne :

```
f = open("exemple.txt", "r")
for ligne in f:
    print(ligne.strip()) # strip() enlève les sauts de ligne
f.close()
```

Lire le contenu d'un fichier texte

c) Lire toutes les lignes dans une liste :

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

Utiliser le mot-clé with

Le mot-clé with ferme automatiquement le fichier, même en cas d'erreur.

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

pas besoin d'appeler f.close().

Exemple complet : lecture et écriture

```
# Écriture
with open("notes.txt", "w") as f:
    f.write("Ali : 15\n")
    f.write("Sara : 18\n")
    f.write("Omar : 12\n")

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

Manipulation de fichiers CSV (Comma Separated Values)

Un fichier CSV contient des données sous forme de tableau, séparées par des virgules (ou des points-virgules).

Exemple de fichier etudiants.csv :

nom	age	note
Ali	20	15
Sara	21	18
Omar	19	12

Lecture sous forme de dictionnaire

```
import csv

with open("etudiants.csv", "r") as f:
    lecteur = csv.DictReader(f)
    for ligne in lecteur:
        print(ligne["nom"], "a eu", ligne["note"])
```

Écriture dans un fichier CSV

```
import csv

with open("nouveaux_etudiants.csv", "w", newline="") as f:
    champs = ["nom", "age", "note"]
    ecrivain = csv.DictWriter(f, fieldnames=champs)
    ecrivain.writeheader() # écrit la première ligne
    ecrivain.writerow({"nom": "Rania", "age": 22, "note": 16})
    ecrivain.writerow({"nom": "Youssef", "age": 20, "note": 14})
```

Manipulation de fichiers JSON (JavaScript Object Notation)

Le JSON est un format très utilisé pour échanger des données entre applications (comme les APIs web).

Exemple de fichier etudiant.json :

```
{  
    "nom": "Ali",  
    "age": 20,  
    "note": 15  
}
```

Lire un fichier JSON

```
import json

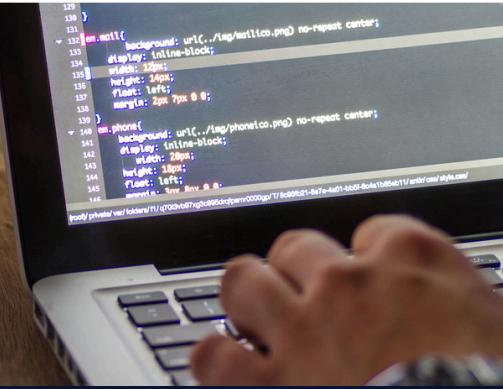
with open("etudiant.json", "r") as f:
    data = json.load(f)
    print(data)
```

Exercices



Exercice :

1. Crée un fichier `produits.csv` avec les colonnes : nom, prix, quantite.
2. Ajoute 3 produits dans le fichier.
3. Lis le fichier et affiche le total de chaque produit (`prix * quantite`).
4. Enregistre le résultat dans un fichier `totaux.json`.



Thank You

