Manipulation des fichiers JSON en Python

Table des matières

1. Cours : Manipulation des fichiers JSON en Python....1
a. Introduction à JSON....1
b. types et structures de données utilisées dans le format json....1
c. Module json de Python....2
d. Convertir un dictionnaire Python en JSON....2
e. Écrire un fichier JSON....2
f. Lire un fichier JSON....2
g. Convertir une chaîne JSON en objet Python....2
h. Erreurs courantes....2
i. Bonnes pratiques....2

1. Cours : Manipulation des fichiers JSON en Python

a. Introduction à JSON

Un **fichier JSON** (JavaScript Object Notation) est un **fichier texte** qui permet de **stocker des données structurées**.

Il est souvent utilisé pour **échanger des données entre des programmes**, ou pour **sauvegarder des informations** (par exemple : des profils d'utilisateurs, des paramètres, des résultats).

Exemple de JSON :

```
{
  "nom": "Alice",
  "age": 30,
  "langages": ["Python", "JavaScript"]
}
```

b. types et structures de données utilisées dans le format json

JSON	Python équivalent
Object	`dict` (dictionnaire)
Array	`list` (liste)
String	`str` (chaîne)
Number	`int` / `float`
Boolean	`True` / `False`
null	`None`

c. Module `json` de Python

Python inclut un module natif appelé `json`.

```
import json
```

d. Convertir un dictionnaire Python en JSON

```
data = {
    "nom": "Alice",
    "age": 30,
    "langages": ["Python", "JavaScript"]
}

# conversion du dictionnaire en objet de type string au format json
json_string = json.dumps(data)
print(json_string)
```

e. Écrire un fichier JSON

On devrait dire : écrire un fichier texte structuré stockant des données au format json.

```
with open("donnees.json", "w") as f:
    json.dump(data, f, indent=4)
```

f. Lire un fichier JSON

On devrait dire : lire un fichier texte structuré stockant des données au format json.

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

g. Convertir une chaîne JSON en objet Python

Une chaîne json est un objet de type string.

```
json_str = '{"nom": "Bob", "age": 25}'
data = json.loads(json_str)
print(data["nom"])  # Bob
```

h. Erreurs courantes

- SyntaxError : mauvaise structure du fichier JSON
- FileNotFoundError : fichier inexistant
- json.JSONDecodeError : contenu mal formé

i. Bonnes pratiques

- Utiliser `indent` pour un fichier lisible
- Gérer les exceptions avec `try...except`
- Toujours fermer les fichiers (ou utiliser `with`)