

RECHERCHE A FAIRE

Format CSV:

- Le format CSV (Comma-Separated Values) est un format de fichier utilisé pour stocker des données tabulaires sous forme de texte.
- Les éléments d'un fichier CSV sont organisés en lignes et en colonnes, avec chaque ligne représentant un enregistrement et chaque colonne représentant un champ de données.
- Les éléments dans un fichier CSV ne sont pas typés, et les commentaires peuvent être ajoutés en utilisant le caractère dièse (#).
- Les outils de manipulation de fichiers CSV incluent des bibliothèques de programmation comme Pandas et NumPy en Python, ainsi que des logiciels spécialisés comme Excel et LibreOffice Calc.
- Les fichiers CSV peuvent être affichés dans un navigateur en utilisant des bibliothèques JavaScript comme D3.js ou en les convertissant en HTML.

Exemple de fichier CSV :

Nom, Age, Sexe

Ibrahima, 34, M

Gabar, 32, M

Marie-Line, 33, F

Cira,35,F

Assane,20,M

Format JSON :

- JSON (JavaScript Object Notation) est un format de données léger et facile à lire et à écrire.
- Les éléments d'un fichier JSON sont organisés en paires clé-valeur, avec chaque paire représentant une propriété et une valeur de données.
- Les éléments dans un fichier JSON peuvent être typés (chaîne, nombre, booléen, etc.) et les commentaires ne sont pas autorisés.
- Les outils de manipulation de fichiers JSON incluent des bibliothèques de programmation comme json en Python, ainsi que des logiciels spécialisés comme MongoDB et CouchDB.

- Les fichiers JSON peuvent être affichés dans un navigateur en utilisant des bibliothèques JavaScript comme jQuery ou AngularJS.

Exemple de fichier JSON :

```
{  
  "nom": "Ibrahima",  
  "âge": 34,  
  "sexe": "M"  
}
```

Format YAML :

- YAML (YAML Ain't Markup Language) est un format de données structuré et facile à lire.
- Les éléments d'un fichier YAML sont organisés en indentations et en sauts de ligne, avec chaque élément représentant un objet ou une liste de données.
- Les éléments dans un fichier YAML peuvent être typés (chaîne, nombre, booléen, etc.) et les commentaires peuvent être ajoutés en utilisant le caractère dièse (#).
- Les outils de manipulation de fichiers YAML incluent des bibliothèques de programmation comme PyYAML en Python, ainsi que des logiciels spécialisés comme Ansible et Kubernetes.
- Les fichiers YAML peuvent être affichés dans un navigateur en les convertissant en HTML.

Exemple de fichier YAML :

```
nom: Gabar  
âge: 29  
sexe: M
```

Manipulation avec Python

CSV, JSON et YAML peuvent être facilement manipulés en Python à l'aide des bibliothèques correspondantes. Voici des exemples de code Python pour lire et écrire dans chaque format:

CSV

```
import csv

# Lecture d'un fichier CSV
with open('fichier.csv') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        print(row)

# Ecriture dans un fichier CSV
with open('fichier.csv', 'w') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(['colonne 1', 'colonne 2', 'colonne 3'])
    writer.writerow(['valeur 1', 'valeur 2', 'valeur 3'])
```

JSON

```
import json

# Lecture d'un fichier JSON
with open('fichier.json') as jsonfile:
    data = json.load(jsonfile)
    print(data)

# Ecriture dans un fichier JSON
with open('fichier.json', 'w') as jsonfile:
    data = {'colonne 1': 'valeur 1', 'colonne 2': 'valeur 2', 'colonne 3': 'valeur 3'}
```

Format XML :

- XML (Extensible Markup Language) est un format de données structuré et extensible utilisé pour stocker et échanger des données entre différents systèmes.
- Les éléments de base dans un fichier XML sont les balises, les attributs et le contenu.
- La structure d'un document XML comprend une déclaration XML, une racine d'élément et des éléments enfants imbriqués.
- Un document XML bien formé doit respecter les règles de syntaxe et de balisage XML, telles que la fermeture des balises et la correspondance des noms de balise.
- Les fichiers XML peuvent être validés par une DTD (Document Type Definition)

Les éléments de base XML :

- Les éléments de base dans un document XML sont les balises, les attributs et le contenu.
- Les balises XML encadrent le contenu et définissent le type d'élément, par exemple **<person>**.
- Les attributs XML sont utilisés pour décrire des caractéristiques spécifiques d'un élément, par exemple **id="123"**.
- Le contenu d'un élément peut être du texte ou d'autres éléments XML imbriqués.

Exemple :

```
<person id="123">
  <name>John Doe</name>
  <age>25</age>
</person>
```

Structure d'un document XML :

- La structure d'un document XML est composée de plusieurs éléments imbriqués les uns dans les autres.
- La première ligne du document XML doit être une déclaration XML qui spécifie la version et l'encodage du document.
- Le document XML doit avoir une racine d'élément unique qui contient tous les autres éléments.

Exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
  <book>
    <title>Harry Potter and the Philosopher's Stone</title>
    <author>J.K. Rowling</author>
    <price>19.99</price>
  </book>
  <book>
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
    <price>14.99</price>
  </book>
</catalog>
```

Document bien formé :

- Un document XML bien formé est un document qui respecte les règles de syntaxe et de balisage XML.
- Les règles incluent la fermeture de toutes les balises, la correspondance des noms de balises et l'encodage des caractères spéciaux.
- Un document XML bien formé peut être validé en utilisant un analyseur XML, tel que **xmllint** en ligne de commande.

Exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <title>Hello, world!</title>
  <content>This is a sample document.</content>
</document>
```

Vérification du document :

- La validation d'un document XML implique de s'assurer qu'il respecte une certaine structure et des règles définies.
- Les documents XML peuvent être validés par une DTD ou un schéma XML.
- La validation peut être effectuée manuellement ou à l'aide d'un outil de validation XML.
- Vérification du document :
- La validation d'un document XML implique de s'assurer qu'il respecte une certaine structure et des règles définies.
- Les documents XML peuvent être validés par une DTD ou un schéma XML.
- La validation peut être effectuée manuellement ou à l'aide d'un outil de validation XML.

DTD (Document Type Definition) :

- Une DTD est un document XML qui spécifie la structure et les règles de validation pour un document XML.
- La DTD définit les types d'éléments, les attributs et les entités.
- La DTD peut être définie de manière interne ou externe.

Exemple de DTD interne :

```
<!DOCTYPE document [
  <!ELEMENT document (title, content)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT content (#PCDATA)>
]>
<document>
  <title>Hello, world!</title>
  <content>This is a sample document.</content>
</document>
```

Exemple de DTD externe :

document.dtd :

```
<!ELEMENT document (title, content)>  
<!ELEMENT title (#PCDATA)>  
<!ELEMENT content (#PCDATA)>
```