

1. Introduction

La transmission d'informations d'une machine à une autre, d'un logiciel à un autre ou d'une base de données à une autre est une tâche récurrente. Le format de données le plus simple est le format CSV. Cela fonctionne bien pour les tableaux mais il ne transmet pas facilement des données non structurées.

Le but de ce TP est de montrer et d'essayer comment manipuler des formats de données non structurés (semi-structurés) avec le langage Python et ses bibliothèques.

2. Ensembles de données de laboratoire :

The **Iris** Dataset. Il s'agit peut-être de la base de données la plus connue de la littérature sur la reconnaissance de formes. Cet ensemble de données se compose de 3 types différents de longueur de pétale et de sépale d'iris (Setosa, Versicolour et Virginica), stockés dans un `numpy.ndarray` de 150x4.

Les lignes sont les échantillons et les colonnes sont : longueur des sépales, largeur des sépales, longueur des pétales et largeur des pétales.

Le tracé ci-dessous utilise les deux premières fonctionnalités. Voir ici pour plus d'informations sur cet ensemble de données.

[https://scikit-](https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html)

[learn.org/stable/auto_examples/datasets/plot_iris_dataset.html](https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html)

Question 1 :

Téléchargez l'ensemble de données et affichez-en une partie.

Indications pour l'utilisation :

- use `sklearn.datasets` library

Question 2:

Écrire les données au format CSV

Indications pour l'utilisation :

- use `StringIO` , `to_csv()`.

Question 3:

Écrire des données au format JSON

Indications pour l'utilisation :

- use `StringIO` , `to_csv()`.

3. XML vers JSON

Q4 : écrivez le code Python qui convertit le fichier XML en code de langage JSON.

Indications pour l'utilisation :

- use `xml.etree.ElementTree` library
- `E.parse()`, `tree.getroot()`..

```
8 <?xml version="1.0" encoding="UTF-8"
9 <?xml-stylesheet type="text/xsl"?>
10 <employees>
11   <details>
12     <firstname>Shiv</firstname>
13     <lastname>Mishra</lastname>
14     <title>Engineer</title>
15     <division>computer</division>
16     <building>301</building>
17     <room>11</room>
18   </details>
19   <details>
20     <firstname>Yuh</firstname>
21     <lastname>Datta</lastname>
22     <title>developer</title>
23     <division>computer</division>
24     <building>303</building>
25     <room>02</room>
26   </details>
27 </employees>
28   <details>
29     <firstname>Rahil</firstname>
30     <lastname>Khan</lastname>
31     <title>Tester</title>
32     <division>computer</division>
33     <building>304</building>
34     <room>10</room>
35   </details>
36   <details>
37     <firstname>Deep</firstname>
38     <lastname>Paraph</lastname>
39     <title>Designer</title>
40     <division>computer</division>
41     <building>305</building>
42     <room>14</room>
43   </details>
44 </employees>
```

4. XML vers JSON

Q5 : écrivez le code Python qui convertit le fichier JSON à côté d'un code de langage XML.

- **Indications pour l'utilisation :**
use `json` library

```
read_activations.py x file.xml x nouveau 5 x data.json x Tnsnames.ora x 2020_json_xml.py x a.xml x
1 {
2   "Name" : "Rani",
3   "Designation" : "PHP Developer",
4   "Salary" : 98000,
5   "Age":27,
6   "Projects" : [
7     {"Topic":"Smart Ambulance","Category":"Android Application","Months":2},
8     {"Topic":"AST","Category":"Embedded System","Months":1},
9     {"Topic":"Plant Nursery","Category":"Website","Months":3}
10  ]
11 }
12
```

Comparison between JSON and XML

JSON	XML
JSON object has a type	XML data is typeless
JSON types: string, number, array, Boolean	All XML data should be string
Data is readily accessible as JSON objects	XML data needs to be parsed
JSON is supported by most browsers	Cross-browser XML parsing can be tricky
JSON has no display capabilities	XML offers the capability to display data because it is a markup language
JSON supports only text and number data type.	XML support various data types such as number, text, images, charts, graphs, etc. It also provides options for transferring the structure or format of the data with actual data.
Retrieving value is easy	Retrieving value is difficult
Supported by many Ajax toolkit	Not fully supported by Ajax toolkit
A fully automated way of deserializing/serializing JavaScript	Developers have to write JavaScript code to serialize/de-serialize from XML
Native support for object	The object has to be express by conventions – mostly missed use of attributes and elements.
It supports only UTF-8 encoding.	It supports various encoding
It doesn't support comments.	It supports comments.
JSON files are easy to read as compared to XML.	XML documents are relatively more difficult to read and interpret.
It does not provide any support for namespaces	It supports namespaces.
It is less secured.	It is more secure than JSON.