

L11 : Traitement de données en tables

1. Organisation de données

De nos jours, nous disposons d'une énorme quantité d'informations. Elles sont stockées dans des formats facilement accessibles et partagés, tels que des emails, des fichiers de type texte ou tableur, des collections d'images, de sons ou de vidéos, ..., etc. Le plus souvent, les données sont organisées sous la forme de **tableaux**.

Par exemple :

Prénom	Date de naissance	Classe	LVB	Spécialité 1
Pierre	19/12/2003	1C	Anglais	NSI
Julie	04/09/2004	1A	Espagnol	Mathématiques
Paul	13/11/2004	1B	Allemand	SES

En informatique, on dit que ces données sont **non structurées**. Ces informations sont représentées ou stockées sans format prédéfini et sont principalement destinées à être interprétées par des humains.

2. Données structurées

a. Cas général

Les **données structurées** sont organisées de façon à être traitées automatiquement et efficacement par un logiciel, mais non nécessairement par un humain. Elles sont généralement stockées et affichées en **colonnes et en lignes**, ce qui permet aux algorithmes et autres outils d'exploration des données d'y accéder et de les analyser via une recherche.

On peut utiliser des fichiers pour stocker un ensemble de données liées entre elles. Ces données doivent pouvoir être conservées de manière persistante et structurée afin de pouvoir être exploitées par des programmes et/ou des utilisateurs différents. C'est le rôle de la **base de données**. Ces fichiers peuvent avoir plusieurs formats, le plus connu étant le **format CSV**.

b. Le format CSV

Le format *Comma-Separated Values* (**CSV**) est un format de fichier informatique de type tableur. Chaque champ, ou colonne provenant d'un tableau, est séparé par une **virgule** ou un **point-virgule**.

Le tableau précédent est représenté comme suit :

```
Prénom,Date de naissance,Classe,LV2,Spécialité 1
Pierre,19/12/2003,1C,Anglais,NSI
Julie,04/09/2004,1A,Espagnol,Mathématiques
Paul,13/11/2004,1B,Allemand,SES
```

L'extension de ce type de fichiers est `.csv`

Prénom, Date de naissance, Classe, LV2 et Spécialité 1 sont appelés des **descripteurs** de données.

"Pierre", "Julie" et "Paul" sont les **valeurs** du descripteur "Prénom".

La fonction principale des fichiers **CSV** est de permettre la **portabilité des données** d'un programme à l'autre.

Un fichier `.csv` présente l'avantage d'être au format texte, donc lisible par n'importe quel programme. Réciproquement, le fichier texte au format `.csv` peut être créé par n'importe quel logiciel ou programme, puis importé dans un tableur où il ne reste plus qu'à refaire la mise en page.

3. Le module `csv` sous Python

Le module `csv` de Python permet de **lire et d'écrire** des fichiers `.csv` très facilement.

On reprend le fichier `.csv` du paragraphe précédent qui contient 4 sous-listes de 5 valeurs :

```
Prénom, Date de naissance, Classe, LV2, Spécialité 1
Pierre, 19/12/2003, 1C, Anglais, NSI
Julie, 04/09/2004, 1A, Espagnol, Mathématiques
Paul, 13/11/2004, 1B, Allemand, SES
```

a. Lecture d'un fichier CSV

```
import csv

# On ouvre le fichier source « eleves.csv »
fichier = open("eleves.csv", "r")

# On crée un objet « lecteur »
lecteur = csv.reader(fichier)

# On parcourt chaque ligne du fichier et on les affiche
for row in lecteur:

    print (row)

# On ferme le fichier source « eleves.csv »
fichier.close()
```

b. Ecriture d'un nouveau fichier CSV

```
import csv

# On crée un fichier source « nouveau_fichier.csv »
fichier = open("nouveau_fichier.csv", "w", newline='')

# On crée un objet « écrire »
ecrire = csv.writer(fichier)

# On crée une ligne dans le fichier
ecrire.writerow(["Emma", "15/07/2004", "1D", "Mandarin", "NSI"])

# On ferme le fichier source « nouveau_fichier.csv »
fichier.close()
```

En utilisant le script précédent pour **lire** le contenu du fichier ainsi créé, on obtient l'affichage suivant :

```
['Emma', '15/07/2004', '1D', 'Mandarin', 'NSI']
>>>
```

Remarque :

La fonction `writerows` avec un "s" à la fin, permet d'écrire en une seule fois tout le contenu d'une liste contenant elle-même des sous-listes.

En revanche, pour écrire uniquement le contenu d'une liste, sans sous-liste, il faut utiliser la fonction `writerow` (sans le "s" à la fin).

Rappels sur la fonction `open` :

La fonction `open` de Python, qui permet d'ouvrir un fichier, possède plusieurs attributs dont les principaux sont :

'r' ouvre le fichier en mode lecture (par défaut)

'w' ouvre le fichier en mode écriture, écrasant au préalable les données existantes

'x' crée un nouveau fichier et l'ouvre en mode écriture

'a' ouvre le fichier en mode écriture, ajoutant des données à la fin du fichier

4. Pour aller plus loin...

Il existe de nombreux autres formats pour structurer les données.

Par exemple :

a. Le format XML

Le format **eXtensible Markup Language (XML)** est un format basé sur l'utilisation de balises pour structurer les données. Les balises sont utilisées pour encadrer un contenu : il y a une balise ouvrante et une balise fermante. En tant que méta-langage XML sert à définir des formats informatiques, c'est-à-dire des façons de représenter de l'information. Les bonnes caractéristiques d'XML - non-ambiguïté, lisibilité, passivité - en font un très bon candidat pour de très nombreux usages, il est donc utilisé dans des secteurs très variés de l'informatique.

Exemple du tableau au format XML :

```
<?xml version="1.0"?>
<annuaire>
  <personne>
    <nom>HAWKING</nom>
    <prenom>Stephen</prenom>
  </personne>
  <personne>
    <nom>REEVES</nom>
    <prenom>Hubert</prenom>
  </personne>
  <personne>
    <nom>EINSTEIN</nom>
    <prenom>Albert</prenom>
  </personne>
</annuaire>
```

b. Le format JSON

Le format **JavaScript Object Notation (JSON)** est un format plus récent utilisé pour représenter des objets qui dérive de la notation des objets du langage JavaScript. Un document JSON est essentiellement un ensemble de paires constituées d'une étiquette et d'une valeur ou d'une liste de valeurs. Les paires sont placées entres accolades et séparées par des virgules. Les valeurs des listes sont placées entre crochets et séparées par des virgules.

Un exemple simple : définition d'un menu (soit un objet composé de membres qui sont un attribut et un tableau lequel contient d'autres objets, les lignes du menu).

<p>Au format JSON :</p> <pre>{ "menu": "Fichier", "commandes": [{ "title": "Nouveau", "action": "CreateDoc" }, { "title": "Ouvrir", "action": "OpenDoc" }, { "title": "Fermer", "action": "CloseDoc" }] }</pre>	<p>L'équivalent en XML :</p> <pre><?xml version="1.0" ?> <root> <menu>Fichier</menu> <commands> <item> <title>Nouveau</value> <action>CreateDoc</action> </item> <item> <title>Ouvrir</value> <action>OpenDoc</action> </item> <item> <title>Fermer</value> <action>CloseDoc</action> </item> </commands> </root></pre>
---	---