Données structurées en tables

Le format CSV

Objectifs :

- **Comprendre ce que sont des données structurées**
- **Comprendre le format CSV**
- **Ouvrir et lire un fichier CSV en Python**

I Données structurées

On trouve de nombreuses données sur Internet en libre accès. Une partie de ses données sont publiques, libres de droit et de réutilisation. C'est ce qu'on appelle l' « open data ».

À faire vous-même 1

Aller sur le site data.gouv.fr puis cliquer sur « documentation ». Résumer en quelques lignes ce que vous avez appris sur l' « open data ».

À faire vous-même 2

- Naviguez sur le site data.gouv.fr et utilisez son moteur de recherche pour découvrir le type de données qu'il contient.
- Recherchez « Services d'attente en gare ». Vous devez constater que les données sont disponibles au format CSV.
  Cliquez sur « prévisualiser » pour voir les premières lignes du fichier.
  Quelle gare met le plus de pianos à disposition ?
  Quelle gare met le plus de « Power&Station » ?
  À votre avis à quoi correspond la première colonne qui a pour entête « UIC » ?

Le format CSV est un format courant pour partager des données. Wikipédia nous dit :

« **Comma-Separated Value**s, connu sous le sigle **CSV**, est un format texte *ouvert* représentant des données tabulaires sous forme de valeurs séparées par des virgules. […]

Un fichier CSV est un fichier texte, par opposition aux formats dits « binaires ». Chaque ligne du texte correspond à une ligne du tableau et les virgules correspondent aux séparations entre les colonnes. Les portions de texte séparées par une virgule correspondent ainsi aux contenus des cellules du tableau. »

Source : https://fr.wikipedia.org/wiki/Comma-separated_values

Voici les premières lignes du fichier « Service d'attente en gare » :

```
UIC;Gare;Piano;Power&Station;Baby-Foot;Distr Histoires Courtes;total
0087741132;Aix-les-Bains le Revard;1;0;0;0;1
0087300822;Belfort - Montbéliard TGV;1;3;0;0;4
0087171926;Champagne-Ardenne TGV;1;0;1;1;3
0087384008;Paris Saint-Lazare;1;0;0;1;2
```

On constate que chaque ligne correspond à une gare différente et que sur une ligne les données sont séparées par des point-virgules.

/! En France la virgule servant comme séparateur décimal on utilise plutôt le point-virgule pour séparer les données.

On peut relativement facilement lire que :

- La gare d'Aix-les-Bains le Revard propose un piano.
- Celle de Belfort – Montbéliard TGV un piano, trois « Power&Station »
- Celle de Champagne-Ardenne TGV un piano, un Baby-Foot et un distributeur d'histoires courtes
- Et celle de Paris Saint-Lazare un piano est un distributeur d'histoires courtes.

IUC, Gare, Piano, Baby-Foot, Distr Histoires Courtes et total sont appelés des **descripteurs.**

« Belfort – Montbéliard TGV » et « Paris Saint-Lazare » sont des **valeurs** du descripteur Gare.

À faire vous-même 3

Télécharger le fichier CSV et ouvrez-le avec un tableur.

Vous devriez obtenir un tableau dont les premières lignes ressemblent à celles-ci :

UIC	Gare	Piano	Power&Station	Baby-Foot	Distr Histoires Courtes
87741132	Aix-les-Bains le Revard	1	0	0	0
87300822	Belfort - Montbéliard TGV	1	3	0	0
87171926	Champagne-Ardenne TGV	1	0	1	1
87384008	Paris Saint-Lazare	1	0	0	1

On peut constater que cela est beaucoup plus lisible ainsi.

II Lecture avec Python

Le fichier ouvert avec le tableur est lisible, malgré cela il n'est pas si facile, sans le manipuler, de déterminer qu'elles sont les gares où il y a exactement deux pianos ou celles où il y a un Baby-Foot.

A Ouvrir un fichier CSV

Le module csv permet de manipuler facilement les fichiers au format CSV.

À faire vous-même 4

/! On suppose que le programme est enregistré dans le même répertoire que le fichier « gares-pianos.csv ».

Recopiez le programme suivant votre éditeur Python et exécute-le.

```
import csv

with open("gares-pianos.csv","r", encoding ="utf -8") as csvfile:
    lecteur = csv.reader(csvfile , delimiter =";")
    for ligne in lecteur:
        print(ligne)
```

Ce programme doit afficher les lignes du fichier les unes à la suite des autres.

La première ligne importe le module csv. La seconde ouvre le fichier « gares-pianos.csv » en lecture (« r » comme read en anglais) et que les caractères sont encodés en utf-8. S'il n'y a pas d'erreurs, cela crée un objet nommé csvfile.

À la ligne suivante, la fonction csv.reader qui prend en argument l'objet csvfile et précise que les données sont séparées par des point-virgules pour créer un objet lecteur.

La boucle affiche sous forme d'une liste de chaînes de caractères chacune des lignes du fichier.

À faire vous-même 5

Dans le programme précédent remplacer dans la dernière ligne print(ligne) par print(ligne[1]).

Qu'est-ce qui est alors affiché ?

Quelle modification faudrait-il faire pour afficher le nom de la gare et le nombre de Baby-Foot ?

B Créer un tableau contenant les données.

On peut créer une liste de listes contenant l'ensemble des données du fichier.

À faire vous-même 6

Recopier puis exécuter le programme ci-dessous.

```
lignes = []
with open( "gares-pianos.csv" ,"r", newline ="", encoding ="utf -8") as csvfile :
    lecteur = csv.reader ( csvfile , delimiter =";")
    for enreg in lecteur :
        lignes.append (enreg)
### Affiche la ligne d'entête
print(lignes[0])
### Affiche le nom de la première gare après l'entête.
print(lignes[1][1])
```

On veut maintenant ***préparer les données***, c'est-à-dire enlever les données inutiles comme la ligne d'entête et la ligne contenant les totaux et convertir les données numériques en entier. Il n'y a pas de données aberrantes dans ce jeu de données mais si cela avait été le cas il aurait fallu décider du traitement qu'il aurait fallu leur faire.

À faire vous-même 7

Modifier le programme de la partie à faire vous-même 6 afin que les données soient préparées.

On veut afficher uniquement les valeurs des gares où un Baby-Foot est à disposition. Le nombre de Baby-Foot est dans la cinquième colonne c'est-à-dire la colonne d'indice 4.

À faire vous-même 8

Compléter le code suivant. On suppose qu'il est écrit après le code précédent et donc que lignes contient le contenu du fichier csv.

```
print("Gare avec un Baby-foot")
for gare in lignes:
    if # À compléter
        print(gare)
```

À faire vous-même 9

Afficher le pourcentage de gare mettant à disposition un Baby-foot. On pourra utiliser la fonction len qui renvoie la taille d'un tableau.

C Créer une liste de dictionnaires contenant les données

La fonction DictReader du module csv permet pour chaque ligne de faire correspondre les descripteurs avec les valeurs correspondantes. Cette fonction se sert de la première ligne pour connaître le nom des descripteurs puis lit toutes les lignes.

À faire vous-même 10

Recopier puis exécuter ce programme. Ce programme lit les données et les prépare.

```
lignes = []
with open( "gares-pianos.csv" ,"r", encoding ="utf -8") as csvfile :
    lecteur = csv.DictReader( csvfile , delimiter =";")
    for ligne in lecteur :
        if  ligne['UIC'] != '':
            ligne['Piano'] = int(ligne['Piano'])
            ligne['Power&Station'] = int(ligne['Power&Station'])
            ligne['Baby-Foot'] = int(ligne['Baby-Foot'])
            ligne['Distr Histoires Courtes'] = int(ligne['Distr Histoires Courtes'])
            ligne['total'] = int(ligne['total'])
            lignes.append (ligne)
### Affiche le dictionnaire correspondant à la première gare.
print(lignes[0])
### Affiche le nom de la deuxième gare
print(lignes[1]['Gare'])
```

Vous devriez avoir pour affichage :

```
OrderedDict([('UIC', '0087741132'), ('Gare', 'Aix-les-Bains le Revard'), ('Piano', 1),
('Power&Station', 0), ('Baby-Foot', 0), ('Distr Histoires Courtes', 0), ('total', 1)])
Belfort - Montbéliard TGV
```

À faire vous-même 11

À l'aide de la variable lignes créée dans le programme précédent, afficher uniquement les gares ayant deux pianos ou plus.

Votre programme devrait afficher trois gares.

À faire vous-même 12

À l'aide de la variable lignes créée dans le « À faire vous-même 10 » afficher uniquement les gares ayant au moins un piano et un Baby-Foot.

Votre programme devrait afficher 14 gares.