

Activité 13b : L'Open Data, des données accessibles à tous

La diffusion des données doit respecter un certains nombres de règles.

- Les données doivent être documentées (les données des données = métadonnées)
- Les données doivent respecter une sémantique (exemple de la date : quel calendrier sert de repère).
- Les données doivent respecter une syntaxe.

Pour assurer l'interopérabilité, les données sont généralement accessibles par le téléchargement d'un fichier de type CSV.

En quoi consiste le format CSV ? Comment traiter les données d'un fichier CSV ?

1. Les fichiers CSV

Définition

Le format CSV (Comma Separated Values) en français, « valeurs séparées par des virgules ». Il s'agit d'un format de fichiers ouvert qui permet de stocker les données d'un tableau. Chaque ligne du fichier correspond à une ligne du tableau alors que les colonnes sont en général séparées par des virgules. On peut très bien remplacer les virgules par des tabulations ou tout autre caractère. Ce format est utilisé pour échanger de manière interopérable des données de tableurs, bases de données, annuaires, etc. entre logiciels différents et/ou plateformes différentes.

Création un fichier CSV

1. **Lancer** Open Calc et **saisir** les données comme ci-contre.
2. **Enregistrer** au format csv (liste.csv)
3. **Ouvrir** le fichier créé avec notepad++
4. **Reporter** sur votre document réponse et **commenter** le résultat

	A	B	C	D
1	NOM	PRENOM	CLASSE	SEXE
2	Lepetit	Alexandre	TS1	G
3	Gaumont	Clemence	TS1	F
4	Guerin	Sophie	TS2	F
5	Kerhuel	Alan	TS2	G
6				

2. Exploitation des fichiers CSV sous Python

Le chargement en mémoire d'un fichier CSV afin d'effectuer un traitement automatique sous python, peut être effectué de façon simplifiée avec la bibliothèque CSV. Pour exemple, le programme suivant permet de charger en mémoire le fichier liste.csv créé précédemment.

```
### Programme de chargement d un fichier csv dans une variable (variable table) ###
import csv

file = open("liste.csv")
table = list(csv.DictReader(file, delimiter=","))
file.close()
```

Programme 1 : Chargement d'un fichier CSV dans une variable



A l'aide de la documentation Python :

5. Décrire le fonctionnement des instructions `open()` » et `close()`

Les fonctions `open()` et `close()` permettent d'ouvrir / fermer un fichier. A l'ouverture le contenu du fichier est chargé dans un objet manipulable ensuite avec d'autres fonctions.

6. Lancer l'exécution de ce programme dans la console Python. **Reporter** sur votre document réponse le contenu de la variable `table` et **commenter** le résultat.

Contenu de `table` :

```
[{'NOM': 'Lepetit', 'PRENOM': 'Alexandre', 'CLASSE': 'TS1', 'SEXE': 'G'}, {'NOM': 'Gaumont', 'PRENOM': 'Clemence', 'CLASSE': 'TS1', 'SEXE': 'F'}, {'NOM': 'Guerin', 'PRENOM': 'Sophie', 'CLASSE': 'TS2', 'SEXE': 'F'}, {'NOM': 'Kerhuel', 'PRENOM': 'Alan', 'CLASSE': 'TS2', 'SEXE': 'G'}]
```

Table contient l'ensemble des données contenues dans le fichier `liste.csv`. Cette donnée `table` est structurée comme une liste de dictionnaires (liste contenant plusieurs dictionnaires).

7. Manipuler dans la console la variable `table` afin de **compléter** le tableau suivant.

Commande exécutée	<code>table[1]</code>	<code>table[2]['PRENOM']</code>	<code>Table[0]</code>	<code>table[0].keys()</code>	<code>table[3]['NOM']</code>
Résultat obtenu	<pre>{'NOM': 'Gaumont', 'PRENOM': 'Clemence', 'CLASSE': 'TS1', 'SEXE': 'F'}</pre>	'Sophie'	<pre>{'NOM': 'Lepetit', 'PRENOM': 'Alexandre', 'CLASSE': 'TS1', 'SEXE': 'G'}</pre>	<pre>dict_keys(['NOM', 'PRENOM', 'CLASSE', 'SEXE'])</pre>	'Kerhuel'
Action réalisée	Contenu de la ligne 2 du tableau (Gaumont Clémence ...)	Valeur « Sophie »	Contenu de la ligne 1 du tableau	Liste des titres des colonnes (NOM, PRENOM, CLASSE SEXE)	Valeur « Kerhuel »

3. Traitement des données en table de l'openData de Bordeaux

Objectif : Traiter des données issues d'un fichier CSV

4. Télécharger le jeu de données répertoriant les parking publics de Bordeaux en cliquant [ici](#)

5. Ouvrir ce jeu avec Notepad++ puis **indiquer** le caractère utilisé pour séparer les lignes et celui séparant les colonnes.

Le délimiteur est un ;

6. Modifier le programme 1 ci-dessus afin de charger dans la variable `table`, le tableau contenu dans le fichier `bor_sigparkpub.csv`

7. Définir les instructions permettant d'extraire de ce fichier le nom du 1^{er} parking répertorié ainsi que le nombre de places disponibles.

Nom du 1^{er} parking :

```
table[0]['nom']
```

Nombre de places :

```
table[0]['nombre_de_places']
```

8. Etablir et **tester** un programme capable d'afficher la liste des noms des parkings publics avec le nombre de places offertes par parking :



gymnase chartrons : 18
argonne : 23
Hopital des Enfants : 30
pont st emilion dessous : 26
pugs bma : 60
achard : 86
Leberthon : 7
abadie : 141
ornano : 26
grand lebrun : 39
Marc Nouaux : 68
Euromaster : 42
barreyre : 30
pont st jean :
faure : 20
Alsace-Lorraine : 128
nicolas beaujon : 39
voie sur berge : 53

Bonus : Etablir un programme capable de répertorier le nombre de place total, gratuites et payantes disponibles sur Bordeaux

