

TD3 : Compteurs de passages de vélos Rennes Métropole

Ce sujet traite des fichiers JavaScript Object Notation (JSON). Ce format permet de stocker des données structurées, par exemple avec une organisation hiérarchique.

Préambule

1. Dans votre dossier PythonOpenData, créez un sous-répertoire TD3 dans lequel vous réaliserez l'ensemble des travaux.
2. Lancer l'éditeur Visual Studio Code.
3. Dans Visual Studio Code, ouvrir le dossier PythonOpenData/TD3.

Ce TD se concentre sur un jeu de données de comptage du nombre de passages de vélos à Rennes au niveau d'un certain nombre de bornes de comptage. Ce jeu de données est issu de la plateforme Open Data de Rennes Métropole, et l'on s'est restreint ici au mois de Juin 2022.

Exercice 1 : Chargement des données

1. Sur CURSUS, télécharger le jeu de données `eco-counter-data_clean.json` et l'enregistrer dans le dossier PythonOpenData/TD3.
2. Visualiser avec Visual Studio Code le contenu du fichier JSON.
3. Quels sont les attributs (clés de dictionnaires) de ce jeu de données ? Que contient l'attribut `date` ?
4. Noter quelques uns des noms de compteurs (attribut `name`).

Dans la suite de ce sujet, on nomme « enregistrement » chaque dictionnaire issu du fichier JSON. Pour information, une date codée sous la forme :

```
{
  "year": 2022,
  "month": 6,
  "day": 3,
  "hour": 20
}
```

correspond au créneau 20h-21h le 3 Juin 2022 et l'enregistrement correspondant indique donc le nombre de passages de vélos au niveau de la borne de comptage considérée sur ce créneau.

Exercice 2 : Extraction d'informations élémentaires

Les questions suivantes nécessitent de manipuler les données depuis Python et le code produit devra être inclus dans le fichier `td3.py`.

1. Charger le contenu du fichier `eco-counter-data_clean.json` dans une liste de dictionnaires, chaque dictionnaire représentant un enregistrement.
2. Écrire une fonction qui retourne une liste sans doublon de tous les noms de compteurs (attribut `name`) contenus dans le jeu de données.

3. Écrire une fonction qui prend en entrée le jeu de données et la liste des noms de compteur et affiche, pour chaque compteur (*i.e.* pour chaque valeur différente de l'attribut `name`), le nombre total de passages de vélos.
4. Écrire une fonction qui prend en entrée le jeu de données et la liste des noms de compteur et affiche, pour chaque compteur, le nombre total de passages de vélos en date du 3 Juin 2022.
5. Écrire une fonction qui prend en entrée le jeu de données et retourne, dans une structure adaptée, le nombre de passages de vélo pour chaque heure de la journée (0h-1h, 1h-2h, ..., 23h-24h), tous compteurs et tous jours confondus.
6. À quelle heure du jour compte-t-on le plus grand nombre de passages de vélos tous compteurs et tous jours confondus ?

Exercice 3 : Export au format CSV

Dans cette partie, il va s'agir d'enregistrer un export du contenu du fichier `eco-counter-data_clean.json` au format CSV. Toutefois, le format CSV n'étant pas adapté pour stocker des listes de valeurs, il faudra pré-traiter les données (et notamment les attributs `gps_coords` et `date`) pour coller aux besoins du format CSV.

1. Écrire une fonction qui prend en entrée le jeu de données précédent et retourne une nouvelle version de ce jeu de données dans lequel l'attribut `gps_coords` a été remplacé par deux nouveaux attributs `latitude` et `longitude` (qui correspondent respectivement au premier et second éléments de la liste `gps_coords`).
2. Écrire une fonction qui prend en entrée le jeu de données précédent et retourne une nouvelle version de ce jeu de données dans lequel l'attribut `date` est représenté par une chaîne de caractères utilisant le format ISO.
3. En utilisant les deux fonctions précédentes et la classe `DictWriter` du module `csv`¹, exporter le jeu de données au format CSV dans un fichier nommé `eco-counter-data_clean.csv`

1. <https://docs.python.org/3/library/csv.html#csv.DictWriter>