

## TD3 : Compteurs de passages de vélos Rennes Métropole

### 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 depuis Anaconda Navigator.
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 nombres 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 d'avril 2024. Les données sont fournies au format JSON (JavaScript Object Notation).

### 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. Notons que chaque enregistrement contient une date. Ainsi, une date codée sous la forme :

```
"date": {  
    "year": 2024,  
    "month": 4,  
    "day": 2,  
    "hour": 10  
}
```

correspond au créneau 10h-11h le 2 avril 2024 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 variable `donnees`. `donnees` doit être une liste des enregistrements, chaque enregistrement étant stocké dans un dictionnaire.
2. Écrire une fonction qui prend en entree la liste des enregistrements et 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 9 avril 2024.
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. Afficher l'heure du jour à laquelle on compte le plus grand nombre de passages de vélos, tous compteurs et tous jours confondus.

### Exercice 3 : Export au format JSON

Dans cette partie, il va s'agir d'enregistrer un export du contenu du fichier `eco-counter-data_clean.json` avec des données corrigées.

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 de la forme `20/03/2007 17h-18h`.
3. En utilisant les deux fonctions précédentes, exporter cette nouvelle version du jeu de données dans un fichier nommé `eco-counter-data_out.json`. Pour une meilleure lisibilité, vous ferez en sorte de fixer l'indentation du fichier obtenu à 2 caractères.