

## TD5 : Dénivelés de randonnées

Ce sujet est une synthèse des travaux avec GraphHopper et avec Json.

Le fichier `mini-rando-gps.json` fournit des séries de positions GPS correspondant à des traces GPS de sorties randonnée de M. Toulemonde. On cherchera dans ce TD à écrire un programme calculant les dénivelés cumulés positif et négatif de chacune de ces randonnées. Pour cela, vous utiliserez l'API GraphHopper via le module Python `graphh`.

### Préambule

1. Dans votre dossier `PythonOpenData`, créez un sous-répertoire `TD5` dans lequel vous réaliserez l'ensemble des travaux.
2. Lancer l'éditeur Visual Studio Code.
3. Dans Visual Studio Code, ouvrir le dossier `PythonOpenData/TD5`.
4. Télécharger le fichier `mini-rando-gps.json` dans le dossier `PythonOpenData/TD5`.
5. Copier le fichier `credentials.json` du TD1 dans le dossier `PythonOpenData/TD5`.
6. Créer un fichier `td5.py` dans lequel vous écrirez le code du TD.

### Exercice 1 : Altitudes avec GraphHopper

1. Écrire une fonction qui prend en entrée un lieu et un client GraphHopper, et affiche l'altitude de ce lieu. Tester en affichant l'altitude de Rennes, de Saint-Malo puis de Chamonix.
2. Ecrire une fonction qui prend en entrée une liste de positions GPS (chacune codée sous la forme d'un dictionnaire comme précisé plus bas) et un client GraphHopper et retourne une liste d'altitudes. Vous pourrez utiliser l'exemple suivant pour vos tests :

```
lst_gps = [  
    {"lng": -1.426533, "lat": 48.005135},  
    {"lng": -1.418127, "lat": 47.986058},  
    {"lng": -1.427611, "lat": 47.989871},  
    {"lng": -1.430202, "lat": 48.000354}  
]
```

### Exercice 2 : Dénivelés

En utilisant la fonction écrite à la question précédente, écrire une fonction qui prend en entrée une liste de positions GPS et un client GraphHopper et retourne la somme des dénivelés positifs (d'une part) et négatifs (d'autre part).

Par exemple, si on a une liste de coordonnées GPS pour lesquelles on a obtenu les altitudes suivantes : `[35.29, 64.36, 48.78, 35.81]` on devrait retourner la paire de valeurs : `(29.07, 28.55)`

### Exercice 3 : Analyse de randonnées

En utilisant la fonction écrite à la question précédente, écrire une fonction qui prend en entrée un nom de fichier JSON (contenant des informations sur diverses randonnées) et un client GraphHopper et affiche, pour chaque randonnée, son nom (attribut "name") et la somme de ses dénivelés positifs (d'une part) et négatifs (d'autre part).

Pour le fichier mini-rando-gps.json, on doit obtenir (après quelque temps, le nombre de requêtes à effectuer étant assez grand) une sortie du type :

```
TraceGPS Le long de la quincampoix - Pire-sur-Seiche
```

```
- Dénivelé positif cumulé : 21.87
```

```
- Dénivelé négatif cumulé : 21.87
```

```
TraceGPS Issued Messac - CIRCUIT DU PORT
```

```
- Dénivelé positif cumulé : 12.71
```

```
- Dénivelé négatif cumulé : 12.71
```

```
TraceGPS Issued Coemes-Retiers
```

```
- Dénivelé positif cumulé : 67.63
```

```
- Dénivelé négatif cumulé : 67.63
```

### Exercice 4 : Fichier de synthèse des randonnées

Écrire une fonction qui prend en entrée un nom de fichier JSON (contenant des informations sur diverses randonnées) et un client GraphHopper, et écrit dans un nouveau fichier JSON (dont le nom sera passé en paramètre de la fonction) une liste de dictionnaires contenant le nom de la randonnée et les informations de dénivelés positif et négatif, soit quelque chose du type :

```
[
  {
    "name": "TraceGPS Le long de la quincampoix - Pire-sur-Seiche",
    "D+": 21.87,
    "D-": 21.87
  },
  {
    "name": "TraceGPS Issued Messac - CIRCUIT DU PORT",
    "D+": 12.71,
    "D-": 12.71
  },
  {
    "name": "TraceGPS Issued Coemes-Retiers ",
    "D+": 67.63,
    "D-": 67.63
  }
]
```