CI 2 – Algorithmique et Programmation

TP – Traitement d'une trace GPS

Remarque

ressources.py

Le fichier ressources.py contient des fonctions permettant d'éviter de re-saisir un certain nombre de fonctions.

Objectif

Les objectif de ce TP sont :
- d'acquérir les données provenant d'un fichier texte (au format kml) ;
- de réaliser des fonctions permettant d'analyser les données pour avoir accès à différentes statistiques.

1 Présentation

L'application «Mes Parcours» disponible sous Android permet, grâce à la puce GPS intégrée à un smartphone, d'enregistrer une trace GPS et d'analyser des données.

Les données y sont enregistrées dans un fichier kml (*Keyhole Markup Language*) dont le formalisme est basé sur le xml (*eXtensible Markup Language*). Le kml est adapté à l'enregistrement de données géographiques.

Exemple

Un fichier kml peut se présenter sous la forme suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
xmlns:gx="http://www.google.com/kml/ext/2.2"
xmlns:atom="http://www.w3.org/2005/Atom">
<Document>
<Style id="track">
<LineStyle><color>7f0000ff</color><width>4</width></LineStyle>
</Style>
<Placemark>
<gx:MultiTrack>
<altitudeMode>absolute</altitudeMode>
<gx:interpolate>1</gx:interpolate>
<gx:Track>
<when>2013-12-23T15:01:50.711Z</when>
<gx:coord>-1.659413 43.331232 152.35414123535156</gx:coord>
<when>2013-12-23T15:01:51.741Z</when>
<gx:coord>-1.659404 43.331358 141.4854278564453</gx:coord>
</gx:Track>
</gx:MultiTrack>
</Placemark>
</Document>
</kml>
```

Les données nous intéressant sont les suivantes :

```
<when>2013-12-23T15:01:50.711Z</when>
<gx:coord>-1.659413 43.331232 152.35414123535156</gx:coord>
```

Exemple

On trouve à l'intérieur des balises <when> et </when> la date et l'heure de la prise de mesure.
À l'intérieur des balises <gx :coord> et </gx :coord> se trouvent la la latitude, la longitude et l'altitude des points de passage.

Question 1

Pour visualiser la trace GPS étudiée, aller sur le site `https://www.google.fr/maps` *dans sa* ***version classique***[1] *et saisir dans la zone de recherche* `http://perso.crans.org/~pessoles/OuCest.kml`.

Dans vos documents personnels, créer un répertoire Informatique dans lequel vous créerez un répertoire 07_GPS. Après avoir créer votre fichier source Python, sauvegardez-le dans ce dossier.
Afin que le répertoire précédent soit considéré comme votre répertoire de travail par Python, saisissez en début de fichier .py les lignes suivantes :

```
from os import chdir
chdir("U:\\Documents\\Informatique\\07_GPS")
```

Le fichier OuCest.py a été enregistré sur le réseau du lycée dans le répertoire Eleves-PTSI.
Si on voulait télécharger ce fichier avec Python (cette méthode n'est pas directement possible sur le réseau du lycée), il faudrait saisir les lignes suivantes :

```
from urllib.request import urlretrieve
urlretrieve("http://perso.crans.org/~pessoles/OuCest.kml", "OuCest.kml")
```

Question 2

Réaliser la fonction Affiche_n qui permet d'afficher les n premières lignes du fichier OuCest.kml. Expliquer pourquoi une ligne vide sépare chacune des lignes.

Une des procédures pour lire un fichier est la suivante :

```
fid = open('fichier.txt','r') # Ouverture du fichier fichier.txt en lecture
file = fid.readlines()        # file contient une liste ou chaque élément représente une ligne du fichier
fid.close()
```

On rappelle qu'une chaîne de caractère peut se manipuler comme un liste.

```
chaine = "abcdefghi"
for i in range(len(chaine)):
    print(chaine[i])
```

Question 3

Réaliser une fonction permettant de savoir si un caractère est compris dans une chaîne de caractères. On nommera la fonction is_char_in_string. Elle prendra comme argument un caractère (le caractère à rechercher) et une chaîne de caractère. Elle renverra une valeur booléenne.

Question 4

Afin d'acquérir les données issues du fichier, il va falloir détecter certains mots clés (par exemple <when>). Réaliser la fonction permettant de savoir si un mot est dans une chaîne de caractère. On nommera la fonction is_word_in_string. On testera notre fonction en prenant une ligne du fichier ainsi que le mot <when>.

1. Si vous êtes dans la nouvelle version, cliquer sur le point d'interrogation en bas à droite de la carte puis cliquer sur *Retour à la version classique de Google Maps*.

Question 5

Réaliser la fonction Affiche_ GPS_ n qui permet d'afficher les n premiers couples de lignes du fichier contenant les balises <when> et <gx :coord>.

Exemple

Le comportement attendu de la fonction est le suivant :

```
Affiche_GPS_n(fichier,1)
    <when>2013-12-23T15:01:50.711Z</when>

    <gx:coord>-1.659413 43.331232 152.35414123535156</gx:coord>
```

Question 6

Une ligne contenant la balise <when> est composée de la date et de l'heure d'une prise de mesure. La fonction ci-dessous permet de renvoyer un tableau composé de l'heure décomposée en heure, minutes et secondes. Expliquer en détail le comportement des lignes 2, 3 4 et 5.

```
1  def heure(ligne):
2      if "when" in ligne :
3          heure = ligne[ligne.find("T")+1:ligne.find("Z")]
4          h = heure.split(":")
5          return [float(h[0]), float(h[1]), float(h[2])]
6      else :
7          return None
```

Question 7

En vous inspirant de la fonction précédente, créer une fonction coordonnée, prenant comme argument une chaîne de caractères et renvoyant un triplet constitué de la longitude, la latitude et de l'altitude d'un point mesuré.

Question 8

Réaliser la fonction permettant d'obtenir un tableau dont chaque élément est une liste [heures,minutes,secondes,longitude,latitude,altitude].

La distance pour aller d'un point à un autre sur une sphère est appelée orthodromie. On note $(\text{lo}_A, \text{la}_A)$ la longitude et la latitude du point A ainsi que $(\text{lo}_B, \text{la}_B)$ la longitude et la latitude du point B. L'orthodromie se calcule alors ainsi en kilomètres :

$$d = 2R \arcsin\sqrt{\sin^2\left(\frac{\text{la}_B - \text{la}_A}{2}\right) + \cos\text{la}_A \cos\text{la}_B \sin^2\left(\frac{\text{lo}_B - \text{lo}_A}{2}\right)}$$

les angles doivent être exprimés en radians. $R = 6371\ km$ représente le rayon de la terre.

Le profil du tracé s'obtient en traçant l'altitude en fonction de la distance parcourue. La fonction orthodromie donnée dans le fichier ressources.py permet de calculer la distance entre deux points prélevés.

Question 9

Écrire la fonction permettant d'obtenir un tableau tab_ traite contenant la liste des triplets [temps écoulés (en s), distances cumulées (en km), altitude(en m)].

Question 10

En utilisant le fichier ressources.py, afficher le profil du parcours.

On peut par exemple mettre les distances cumulées dans un tableau et les altitudes dans un autre tableau et utiliser les lignes suivantes (attention les deux tableaux doivent avoir la même taille) :

```
import matplotlib.pyplot as plt
import numpy as np

p1=plt.plot(distance_cumulee,altitude,linewidth=3)
plt.grid(True, which="both", linestyle="dotted")
plt.show()
```

Question 11

Réaliser une fonction permettant de déterminer le dénivelé positif total.

Question 12

Réaliser une fonction permettant de déterminer la vitesse moyenne du randonneur.

Afin de sauvegarder les données traitées, on souhaite qu'elles soient écrites dans un fichier texte.

Question 13

Écrire une fonction permettant d'écrire le tableau tab_traite dans un fichier texte. Les valeurs seront séparées par des tabulations. On insèrera un retour à la ligne après chaque triplet de données.

Question 14

Écrire une fonction permettant d'écrire le tableau tab_traite dans un fichier texte. En plus de ce qui est demandé dans la question précédente, on souhaite que les nombres soient écrits avec des virgules.

Pour créer et écrire dans un fichier texte, on peut procéder ainsi :

```
fid = open("fichier.txt",'w')
fid.write("champ1 \tab champ2")
fid.close()
```