

Exercice 1 : la localisation sur Terre.

1- Créer une fonction permettant de transformer les coordonnées DD en DMS et vis versa... :_dd_dms et dms_dd

2- Créer une fonction permettant de donner une distance à vol d'oiseau entre des points de surface du globe : distance_vol_oiseau ... :

formule :

$$d = ACOS(SIN(latA) * SIN(latB) + COS(latA) * COS(latB) * COS(longB - longA)) * R.$$

Avec les latitudes et longitudes en radians ; conversion :

angle en radians = (angle en degrés/180)*Pi

R= rayon moyen de la Terre à l'équateur= 6378137m

pensez à importer les modules nécessaires : from math import sin, cos, acos

Exercice 2 : La géolocalisation par système satellites.

1- Donner les coordonnées et l'altitude associées à la trame :

\$GPGGA,174553.258,423600.8900,N,084821.8700,E,1,04,3.1,0.0,M,,,,,0000*0E

2- Construire une fonction python NEMEA_DD qui donne les coordonnées en format DD si l'on rentre une trame \$GPGGA. Facultatif, c'est c....t !!!

Exercice 3 : Création de cartes et de trajets, choix des trajets les plus courts.

1- Créer une carte grâce au module folium : la zone importe peu, je veux 4 points sur la carte reliés par des lignes.

Indispensable :

→ installation de *folium*.

Pip **install** folium

→ début de programme.

Import folium as fol

Attribution de la carte à une variable(c'est un tableau, une image matricielle!).

Code :

Macarte=fol.Map(location=[lat,long], zoom_start=10)

location : centre la carte, coordonnées en DD. → indispensable

zoom_start : permet d'afficher avec un zoom → facultatif

tiles : indique une des 200 cartes de fond appelée, par défaut, openstreetmap → facultatif

divers instructions du module folium :

variable.save('nomde fichier.html') → crée un fichier de type html qui affiche la carte.

Exemple : Macarte.save('MaCarte.html')

fol.Marker([lat,long], popup='nomdupoint').add_to(Macarte) → place des icônes à des endroits précis sur la carte nommée Macarte et affiche nomdupoint quand on clique dessus.

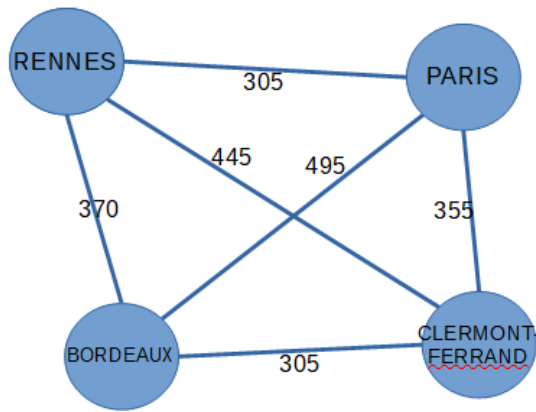
Exemple :

folium.Marker([49.1,0.0], popup="trouPaumé", icon=folium.Icon(color='red')).add_to(Macarte)

folium.PolyLine([(49.1,0),(48.5,0),(48.5,1)],color="blue",weight=2.5, opacity=0.8).add_to(carte) → on trace un trajet bleu de largeur 2,5px et d'opacité 0,8 qui passe par les 3 points aux coordonnées notées.

2- Réaliser le graphe non orienté pondéré des trajets à vol d'oiseau entre ces 4 points. (utilisez la formule ou le programme réalisé en exercice 1)

Remarque : On peut représenter un « graphe » qui représente la disposition des points sur votre carte. En admettant qu'on peut aller de chaque point vers tous les points, un graphe pourrait donner cela :



On trouve sur les lignes de liaison les valeurs de chaque liaison(ici des distances). C'est un graphe non orienté(il n'y a pas de sens obligatoire sur chaque ligne) avec des arrêtes de poids différents.

Cette représentation peut aider à la résolution de problème comme le chemin le plus court entre deux points ou celui nécessaire pour passer par tous les points une seule fois en un minimum de distance.

Projet :

Situation :

Jean-Louis a 15 ans et c'est déjà dur de s'appeler Jean-Louis aujourd'hui !

Il à envie de visiter Paris avec Mireille, sa grand-mère de 87ans(c'est bizarre mais bon!). Elle a du mal à marcher mais c'est la grève des transports(métro, bus, taxi, Uber...), , c'est vraiment la poisse !

En plus, elle veut absolument voir 6 monuments avant de repartir par là où elle arrive demain matin : La gare du Nord. Le monuments sont les suivants :

ARC_TRIOMPHE = [48.873804 , 2.29512]

EIFFEL = [48.858053 , 2.2944991]

JARDIN_DES_PLANTES = [48.843971 , 2.359786]

LOUVRE = [48.860572 , 2.337657]

ORSAY =[48.86004 , 2.326533]

QUAI_BRANLY =[48.860903 , 2.297582]

Et pour la gare du nord :

GARE_NORD = [48.882505 , 2.354784]

Ressources :

- Vous pouvez utiliser les variables monuments telles qu'elles sont présentées ici, ça vous facilitera la tâche

- Vous pouvez créer une liste avec vos variables :

lieux = [ARC_TRIOMPHE, EIFFEL, GARE_NORD, JARDIN_DES_PLANTES, LOUVRE, ORSAY ,
QUAI_BRANLY]

*Rappel : dans la liste 'lieux', lieux[2] correspond aux coordonnées de la gare du nord,
lieux [3][1] correspond à la longitude du jardin des plantes.*

Niveau 1 :

Créer la carte de Paris en format html qui permet de visualiser les 6 endroits du parcours parisien, attribuer un marqueur à chaque endroit avec le nom de l'endroit associé en popup et afficher le trajet en couleur rouge.

Niveau 2 :

Niveau 1 + Utiliser vos fonctions pour évaluer la distance total du parcours.

Niveau 3 :

Trouver le trajet le plus court adapté aux vieilles jambes de Mireille en tâtonnant mais e utilisant votre joli programme.

Niveau Ultime :

Créer un programme qui trouve le chemin à vol d'oiseau le plus court pour parcourir autant de points que l'on veut à Paris.

Le programme doit demander le nom des monuments et les deux coordonnées à l'utilisateur pour chaque monument.