

# TP3

## Les objets

### Exercice 1 :

1.

Ecrire un programme qui permet de saisir plusieurs lieux et qui grâce à google maps permet de récupérer les coordonnées GPS.

Pour cela, il faudra créer une classe Lieu qui aura pour variables :

- nom
- adresse
- latitude
- longitude

Dans le constructeur, vous devrez récupérer les coordonnées grâce à la fonction “ coordgps ” de la classe gmap ci-dessous que vous devrez intégrer dans votre code.

```
import googlemaps
```

```
class gmap:
```

```
    def coordgps(adresse):
        gmaps = googlemaps.Client(key='AIzaSyBY7RDZYuBbD6I8uFPxFdkE55Xx90720b0')
        geocode_result = gmaps.geocode(adresse)
        lat = geocode_result[0]["geometry"]["location"]["lat"]
        lng = geocode_result[0]["geometry"]["location"]["lng"]
        return lat,lng
```

Dans la classe lieu, vous devrez écrire une méthode “ detail “ qui affichera le résultat complet.

### Exercice 2 :

Créer une classe Personne qui permet de définir des objets représentant des personnes.

Une personne est décrite par son nom, son prénom, son âge et son sexe.

Votre classe doit proposer :

1. Un constructeur par défaut qui ne prend aucun paramètre et qui permet de créer le fameux John Doe (on supposera que ce monsieur a 30 ans).
2. Un constructeur qui prend en paramètre toutes les informations (nom, prénom, âge et sexe) et crée l’objet correspondant convenablement.
3. Un ensemble d’accesseurs (ou getters) qui permettent de récupérer les valeurs des différents attributs de l’objet (ex. une méthode getName() qui permet de connaître le nom de la personne et ainsi de suite).
4. Une méthode sameLastName(Personne p) qui prend en paramètre un deuxième objet de type Personne et qui permet de savoir si les deux personnes ont le même nom de famille.
5. Une méthode oldest(Personne p) qui compare la personne appelante avec la personne fournie en paramètre et retourne celle qui est la plus âgée.
6. Dans un fichier main.py, testez la classe et les méthodes.

### Exercice 3 :

Considérons une classe appelée Point ayant les attributs suivants :

- abs : de type float
  - ord : de type float
1. Définissez la classe Point et son constructeur.
  2. Définissez les getters et setters pour les deux attributs en utilisant le décorateur `@property`
  3. Ecrivez la méthode `calculer_distance` → 'Point' qui permet de calculer la distance entre le point de l'objet courant (`self`) et l'objet `p` passé en paramètre.
  4. Pour rappel mathématiques : <http://www.mathematiques-lycee.com/geometrie/2nde-01-longueur-segment.html>
  5. En s'inspirant de la question 3 écrivez la méthode `calculer_milieu`.
  6. Dans un fichier `main.py`, testez toutes les classes et méthodes que vous implémentées.

Considérons maintenant une deuxième classe appelée TroisPoints ayant les attributs suivants :

- premier : un attribut de type point
  - deuxieme : un attribut de type point
  - troisieme : un attribut de type point
1. Définissez les getters / setters ( avec le décorateur `@property` ) et un constructeur acceptant trois paramètres.
  2. Ecrivez une méthode `sont_alignees` → 'bool' qui retourne `TRUE` si les trois points premier, deuxieme, troisieme sont alignés. Nous rappelons que trois points A,B et C sont alignés si  $AB = AC + BC$ ,  $AC = AB + BC$  ou  $BC = AC + AB$  ( AB désigne la distance séparant le point A du point B, pareillement pour AC et BC ) .
  3. Ecrivez une méthode `est_isocèle` → 'bool' qui retourne `TRUE` si les points forment un triangle isocèle. Nous rappelons qu'un triangle ABC est isocèle si  $AB = AC$  ou  $AB = BC$  ou  $BC = AC$ .
  4. Dans un fichier `main.py`, testez toutes les classes et méthodes que vous implémentées.