

Première partie (8 points)**Exercice 1 (4 points)**

On interroge l'API *Découpage administratif communes* du gouvernement français en envoyant une requête à l'adresse :

```
https://geo.api.gouv.fr/communes?nom=Paris&boost=population&fields=population,surface
```

pour récupérer les données géographiques (population et superficie) de la ville de Paris. La réponse de l'API est :

```
[{"population": 2190327, "surface": 10528.81, "nom": "Paris"},
{"population": 959, "surface": 2872.7, "nom": "Parisot"},
{"population": 593, "surface": 2822.99, "nom": "Parisot"},
{"population": 23924, "surface": 843.58, "nom": "Cormeilles-en-Parisis"},
{"population": 11981, "surface": 1078.46, "nom": "Seyssinet-Pariset"},
{"population": 1963, "surface": 1101.44, "nom": "Fontenay-en-Parisis"},
{"population": 298, "surface": 275.77, "nom": "Paris-l'Hôpital"},
{"population": 4244, "surface": 1526.09, "nom": "Le Touquet-Paris-Plage"}]
```

Remarque : Les superficies sont exprimées en hectares, et 1 kilomètre carré correspond à 100 hectares.

1. Comment s'appelle le format dans lequel les données sont reçues ?

On suppose que la réponse de l'API est stockée dans une variable `reponse`.

2. Quel est le type de la variable `reponse` ? Et de l'élément `reponse[0]` ?
3. Quelles expressions permettent :
 - d'accéder à la population de la ville de Paris ?
 - d'accéder à la superficie de la ville de Paris ?
 - de calculer la densité de la ville de Paris ?
4. À quelle adresse faut-il interroger l'API pour récupérer les données géographiques de la ville de Nogent ?

Exercice 2 (4 points)

On donne la définition d'une fonction `openfoodfacts` :

```
def openfoodfacts(code_barres):
    url = f"https://world.openfoodfacts.org/api/v0/product/{code_barres}.json"
    reponse = requests.get(url)
    reponse = reponse.json()
    if reponse['status'] == 1:
        dico = {'nom': reponse['product']['product_name'],
                'nutri_score': reponse['product']['nutrition_grade_fr']}
    else:
        dico = {'nom': 'Produit inconnu'}
    return dico
```

Si on interroge l'API Open Food Facts à partir d'un code-barres erroné, on obtient une réponse du type :

```
{"code": "8076800376998", "status": 0, "status_verbose": "product not found"}
```

1. Dire, en justifiant, ce que renvoie la fonction `openfoodfacts` dans le cas où on l'appelle avec un code-barres erroné en argument.

L'appel `openfoodfacts('3166352967037')` renvoie ceci :

```
{'nom': "Poisson à l'andalouse riz safrané et courgettes grillées",  
'nutri_score': 'a'}
```

2. Écrire la spécification de la fonction `openfoodfacts`.
3. Définir une fonction `nutriscore_a` qui :
 - prend en paramètre d'entrée un tableau `tab_codes` contenant des numéros de code-barres, et qui
 - renvoie, sous forme de tableau, le nom des produits dont le nutriscore est « A ».

Seconde partie (12 points)

Copier sur le bureau le fichier `devoir1.zip` depuis le dossier `Devoir` du réseau, le décompresser, puis ouvrir avec Jupyter le fichier `devoir1.ipynb`.

Lorsque vous aurez terminé, vous renommerez votre fichier `nom_prenom.ipynb` et vous le déposerez dans le dossier `Rendu` du réseau. Attention, une fois déposé dans `Rendu`, votre travail n'est plus modifiable.