

## TD : Les villes où il fait bon vivre

### 1 Préambule

Ce travail se fait par groupes de 2 étudiants (éventuellement 3). Le TP est conçu pour être réalisé en 2 heures lors de la séance de TD. La date limite de rendu (dépôt) précisée sur CURSUS correspond à l'heure de fin du TD.

### 2 Contexte

L'objectif de ce TP est d'appliquer des compétences d'accès aux données (fichiers et APIs), de nettoyage et de croisement, et d'analyse basique pour répondre à une question concrète : *où il fait bon aller demain* parmi les chefs-lieux de région français en conciliant météo et pollution.

Les jeux de données proviennent d'Open Data (INSEE pour les communes/régions) et d'APIs publiques (prévisions météo et pollution). Deux modalités d'accès aux données sont proposées (A : hors-ligne ou B : en ligne) : ne traitez la modalité B que si vous avez le temps.

### 3 Énoncé

À partir de données Open Data, évaluer les chefs-lieux de région français sur trois critères et produire un classement de la *douceur de ville* pour le jour courant.

#### 3.1 Critères (pour chaque ville)

1. **Température actuelle** (idéalement proche de 20 °C).
2. **Niveau de pollution actuel** (concentration PM2.5 ; plus c'est faible, mieux c'est).

#### 3.2 Données à utiliser

**INSEE (fichiers fournis)** Deux fichiers :

- liste des communes avec leurs coordonnées GPS (code INSEE, nom, latitude, longitude) : `communes_gps.json`
- liste des régions avec la colonne `CHEFLIEU` (code INSEE du chef-lieu) : `v_region_2025.csv`

Ces fichiers permettent d'obtenir la liste des chefs-lieux et leurs coordonnées.

**Météo et pollution :**

- **Niveau A (hors-ligne)** : fichiers JSON simulant les réponses d'API pour la météo et la pollution : `villes_pollution.json` et `villes_temperature.json`.
- **Niveau B (en ligne)** : appels réels aux API OpenWeather (Current Weather / Air Pollution) pour récupérer les données actuelles :
  - Page de création d'une clé d'API : [https://home.openweathermap.org/api\\_keys](https://home.openweathermap.org/api_keys) (nécessite d'avoir créé un compte, gratuit)
  - Météo actuelle : <https://openweathermap.org/current>
  - Pollution actuelle (PM2.5) : <https://openweathermap.org/api/air-pollution>

## 4 Niveaux de réalisation

### 4.1 Niveau 1 – Extraction et affichage

- Charger les fichiers INSEE fournis et la source météo/pollution (fichiers locaux ou APIs selon l'approche retenue).
- Pour chaque chef-lieu de région, afficher :
  - la température du moment,
  - le niveau de pollution courant (PM2.5).

### 4.2 Niveau 2 – Notation (étoiles)

- Transformer chaque indicateur en une note sur 5 étoiles.
- Exemple de règle proposée : pour chaque ville, la température et la pollution sont converties en 1 à 5 étoiles selon des seuils fixes.

Les intervalles pour la température sont évalués du plus strict au plus large : la première condition satisfaite donne la note.

Température (°C)	Étoiles
19 – 21	5
18 – 22	4
15 – 25	3
10 – 30	2
autres	1

Pour la pollution (PM2.5 en  $\mu\text{g}/\text{m}^3$ ), les seuils sont non chevauchants et augmentent avec la concentration (plus la pollution est faible, plus la note est élevée) :

PM2.5 ( $\mu\text{g}/\text{m}^3$ )	Étoiles
0 – 1	5
1 – 2	4
2 – 5	3
5 – 10	2
> 10	1

- Afficher un tableau récapitulatif contenant, pour chaque ville, les deux notes (météo, pollution).

### 4.3 Niveau 3 – Classement final

- Combiner les deux critères (somme des étoiles) pour obtenir un score global de "douceur de vivre".
- Afficher la ville la mieux classée (et idéalement le Top 5).

## 5 Travail attendu et livrables

Vous devez produire un **fichier Python** nommé `tp_villes_bon_vivre.py` qui réalise les traitements demandés. Le code doit être clair, structuré en fonctions, et éviter les traitements redondants. Utilisez les structures de données appropriées (listes, dictionnaires). Le fichier Python devra contenir en en-tête, commentés, les noms et numéros étudiants des membres du groupe.

Ne mettez jamais vos clés d'API dans les fichiers déposés. Votre script lira les clés depuis un fichier `credentials.json` (que vous ne déposerez pas), au format suivant :

```
{  
    "OpenWeather": "VOTRE_CLEF_OPENWEATHER",  
}
```