

# Travaux dirigés

## Thème : Python & Cartographie

### *Visualisation de villes françaises avec coordonnées GPS et Folium*

Niveau	Durée	Prérequis	Langage
Lycée / BTS / IUT	2 heures	Python débutant	Python 3

## Objectifs pédagogiques

À l'issue de ce TD, l'étudiant sera capable de :

- Lire et manipuler un fichier CSV avec la bibliothèque pandas
- Rechercher des données dans un DataFrame par nom de ville
- Utiliser la bibliothèque Folium pour générer une carte interactive
- Placer un marqueur géolocalisé sur une carte et enregistrer le résultat en HTML
- Structurer un programme Python avec des fonctions et un bloc main

## Contexte et présentation des fichiers

Dans ce TD, vous disposez de plusieurs fichiers :

Fichier	Description
<b>villes_coo.csv</b>	Base de données de ~39 000 villes françaises avec leurs coordonnées GPS (latitude, longitude). Séparateur point-virgule (;). En-têtes : ville, latitude, longitude.
<b>villes_coordonnees.csv</b>	Fichier enrichi avec des informations administratives (région, département, commune, EPCI...) et les coordonnées géolocalisées.
<b>villes_coo.py</b>	Script Python principal à étudier, compléter et améliorer.
<b>carte.html</b>	Carte interactive générée par le script, affichant un marqueur sur la ville saisie par l'utilisateur.

## Partie 1 — Analyse du script existant

### 1.1 — Lecture du code

Voici le script Python fourni. Lisez-le attentivement avant de répondre aux questions.

```
import pandas as pd
import folium

def obtenir_coordonnees_de_csv(nom_ville, fichier_csv):
    data = pd.read_csv(fichier_csv, sep=';')
    ville_data = data[data['ville'] == nom_ville]
    if not ville_data.empty:
        latitude = ville_data['latitude'].values[0]
        longitude = ville_data['longitude'].values[0]
        return latitude, longitude
    else:
        print(f'Coordonnées non trouvées pour {nom_ville}.')
        return None, None

def main():
    fichier_csv = './villes_coo.csv'
    nom_ville = input('Entrez le nom de la ville (1ère lettre en majuscule) : ')
    latitude, longitude = obtenir_coordonnees_de_csv(nom_ville, fichier_csv)
    if latitude is not None and longitude is not None:
        carte = folium.Map(location=[latitude, longitude], zoom_start=12)
        folium.Marker([latitude, longitude], popup=nom_ville).add_to(carte)
        folium.Marker([latitude, longitude], tooltip=nom_ville).add_to(carte)
        carte.save('carte.html')
        print("La carte a été enregistrée sous le nom 'carte.html'.")
    else:
        print('Coordonnées non trouvées. Vérifiez le nom ou le fichier CSV.')

if __name__ == '__main__':
    main()
```

### 1.2 — Questions de compréhension

Répondez aux questions suivantes en vous appuyant sur le code ci-dessus.

**Question 1 — Quel est le rôle de la ligne `import pandas as pd` ? Pourquoi utilise-t-on l'alias `pd` ?**

Réponse : .....

**Question 2 — À quoi sert le paramètre `sep=';` dans `pd.read_csv()` ?**

Réponse : .....

**Question 3 — Que retourne la fonction `obtenir_coordonnees_de_csv()` si la ville n'est pas trouvée ?**

Réponse : .....

**Question 4 — Quelle est la différence entre `popup` et `tooltip` dans Folium ?**

Réponse : .....

**Question 5 — Pourquoi utilise-t-on le test `if __name__ == '__main__':` à la fin du script ?**

Réponse : .....

## Partie 2 — Exploration du fichier CSV

### 2.1 — Structure du fichier villes\_coo.csv

Ouvrez le fichier **villes\_coo.csv** dans un éditeur ou avec Python. Complétez le tableau suivant :

Information	Valeur observée
Nombre de lignes (hors en-tête)	
Noms des colonnes	
Séparateur utilisé	
Exemple de ville (ligne 2)	
Latitude de cette ville	
Longitude de cette ville	

### 2.2 — Manipulation avec pandas

Écrivez un script Python court (dans la zone ci-dessous) permettant d'afficher les 5 premières lignes du fichier **villes\_coo.csv** et le nombre total de villes répertoriées.

```
# Complétez ce script
import pandas as pd

# Lecture du fichier
data = _____

# Afficher les 5 premières lignes
print(_____)

# Afficher le nombre total de villes
print('Nombre de villes :', _____)
```

## Partie 3 — Exercices de programmation

### Exercice 1 — Recherche insensible à la casse

Le script actuel exige que la première lettre de la ville soit en majuscule (ex : "**A**ngers" et non "**a**ngers"). Modifiez la fonction `obtenir_coordonnees_de_csv()` pour rendre la recherche insensible à la casse.

*Indice : utilisez la méthode `.str.lower()` de pandas.*

```
def obtenir_coordonnees_de_csv(nom_ville, fichier_csv):
    data = pd.read_csv(fichier_csv, sep=';')
    # Modifier la ligne ci-dessous pour ignorer la casse
    ville_data = data[data['ville'].str.lower() == nom_ville.lower()]
    ...
```

## Exercice 2 — Afficher plusieurs villes

Modifiez le programme pour permettre à l'utilisateur d'entrer plusieurs villes (séparées par des virgules) et placez un marqueur pour chacune sur la même carte. Complétez le squelette suivant :

```
def main():
    fichier_csv = './villes_coo.csv'
    saisie = input('Entrez des villes séparées par des virgules : ')
    villes = [v.strip() for v in saisie.split(',')]

    # Initialiser la carte centrée sur la France
    carte = folium.Map(location=[46.5, 2.3], zoom_start=6)

    for nom_ville in villes:
        lat, lon = obtenir_coordonnees_de_csv(nom_ville, fichier_csv)
        if lat is not None:
            # Ajouter un marqueur pour cette ville
            _____

    carte.save('carte_multi.html')
    print('Carte enregistrée sous carte_multi.html')
```

## Exercice 3 — Couleurs personnalisées

Folium permet de personnaliser la couleur des marqueurs via `folium.Icon()`. Modifiez le code de l'exercice 2 pour que chaque marqueur soit affiché dans une couleur différente selon un ordre cyclique (ex : rouge, bleu, vert, orange...).

*Documentation utile : `folium.Icon(color='red', icon='info-sign')`*

### Zone de réponse :

```
couleurs = ['red', 'blue', 'green', 'orange', 'purple']

for i, nom_ville in enumerate(villes):
    lat, lon = obtenir_coordonnees_de_csv(nom_ville, fichier_csv)
    if lat is not None:
        couleur = _____
        folium.Marker(
            [lat, lon],
            popup=nom_ville,
            tooltip=nom_ville,
            icon=folium.Icon(color=_____, icon='info-sign')
        ).add_to(carte)
```

## Partie 4 — Pour aller plus loin

### Exercice 4 — Utilisation de `villes_coordonnees.csv`

Le fichier `villes_coordonnees.csv` contient des informations administratives riches : région, département, EPCI, etc. La colonne de géolocalisation est sous la forme "latitude, longitude" dans une seule cellule.

Écrivez une fonction qui :

1. Charge `villes_coordonnees.csv`
2. Demande à l'utilisateur un nom de département
3. Affiche toutes les communes de ce département sur une carte Folium

*Indice : la colonne géolocalisation contient des valeurs comme "46.153721024, 4.925850148". Utilisez `str.split(',')` pour séparer latitude et longitude.*

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

## Exercice 5 — Questions de réflexion

**Question A — Citez deux limites du script actuel liées à la qualité des données.**

Réponse : .....

.....

**Question B — Quelle bibliothèque alternative à Folium pourrait-on utiliser pour créer des cartes en Python ?**

Réponse : .....

**Question C — Dans quel format est enregistrée la carte générée ? Quels sont les avantages de ce format ?**

Réponse : .....

.....