

# TP — Exploitation d'une API météo et génération d'une page HTML

---

## Objectifs

- interroger une **API REST** en Python
- manipuler des données JSON
- extraire et formater des informations utiles
- générer dynamiquement une **page HTML** à partir de données Python
- comprendre la séparation **données / affichage**

## Prérequis

- bases de Python (**variables**, **listes**, **boucles**)
- notions de fichiers (**open**, **read**, **write**)
- HTML de base (balises simples)

## Contexte

On souhaite créer un petit outil météo pédagogique :

- Python récupère les données météo via l'API **Open-Meteo**
- Les données sont intégrées dans une page HTML
- Le résultat est affiché dans un navigateur

## Organisation du travail

```
tp_meteo_PY_HTML/  
├── meteo.py          # script Python  
└── template.html    # page HTML modèle
```

## Étape 1 — Récupérer les données météo

Utiliser l'API Open-Meteo pour obtenir :

- la température actuelle
- la vitesse du vent
- les prévisions horaires (température + pluie)
- limitées à **2 jours**

## Étape 2 — Extraire les informations utiles

À partir du JSON :

- température actuelle

- vitesse du vent
- les **6 prochaines heures** de prévision

## Étape 3 — Générer dynamiquement une page HTML

Le fichier HTML sert de **modèle**. Python se charge uniquement de remplacer les données dynamiques.

- ouvrir un fichier HTML modèle
- remplacer des balises de type `{{VARIABLE}}`
- écrire le résultat dans un nouveau fichier HTML

## Étape 4 — Visualisation

- ouvrir la page générée dans un navigateur
- vérifier la cohérence des données affichées

## Résultat attendu

Une page HTML affichant :

- la ville
- la température actuelle
- la vitesse du vent
- les prévisions des prochaines heures

---

## Codes (à compléter)

`template.html` (fourni tel quel)

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Météo</title>
</head>
<body>

<h1>Météo – {{CITY}}</h1>

<p>
  – Température : {{TEMP}} °C<br>
  – Vent : {{WIND}} km/h
</p>

<h3>Prochaines heures</h3>
<ul>
  {{FORECAST}}
</ul>
```

```
</body>
</html>
```

### meteo.py (à compléter)

```
import requests

# -----
# Paramètres
# -----

CITY = "Rodez"
LAT = 44.35258
LON = 2.57338
TIMEZONE = "Europe/Paris"

URL = "https://api.open-meteo.com/v1/forecast"

# -----
# Requête API
# -----

params = {
    "latitude": LAT,
    "longitude": LON,
    "current_weather": True,
    "hourly": ["temperature_2m", "precipitation"],
    "forecast_days": 2,
    "timezone": TIMEZONE
}

response = requests.get(URL, params=params)

if response.status_code != 200:
    print("Erreur lors de l'appel à l'API météo")
    exit()

data = response.json()

# -----
# Extraction des données
# -----

current = data["current_weather"]
hourly = data["hourly"]

# Indice : current est un dictionnaire
temperature = ...      # température actuelle
windspeed = ...        # vitesse du vent

# Indice : les données horaires sont des listes
```

```
# Exemple : hourly["time"]
times = ...           # heures (6 prochaines)
temps = ...           # températures
pluie = ...           # précipitations

# -----
# Génération HTML
# -----

with open("template.html", "r", encoding="utf-8") as f:
    html = f.read()

forecast_html = ""

# Boucle pour créer les <li>
# -----
# À compléter
# Exemple attendu :
# <li>10:00 → 12.3 °C | pluie : 0 mm</li>
#
# l'API renvoie : 2026-02-08T10:00
# t[11:16] permet d'extraire l'heure
# -----

html = html.replace("{{CITY}}", CITY)
html = html.replace("{{TEMP}}", str(temperature))
html = html.replace("{{WIND}}", str(windspeed))
html = html.replace("{{FORECAST}}", forecast_html)

with open("meteo_resultat.html", "w", encoding="utf-8") as f:
    f.write(html)

print("Page HTML générée : meteo_resultat.html")
```

---

## Questions

- Comment changer de ville ?
- Comment afficher 12 heures au lieu de 6 ?
- Comment ajouter la pluie totale de la journée ?

### Bonus

- Ajouter une couleur différente si pluie > 0 mm
  - Ajouter la date dans la page HTML
  - Générer une deuxième page pour une autre ville
-