

# Kata 2

## Objectifs pédagogiques

- Définir et expliquer les concepts de base de la programmation asynchrone (callbacks, promesses, async/await)
- Utiliser des callbacks pour gérer des opérations asynchrones
- Utiliser `fetch` pour effectuer des requêtes HTTP et gérer les promesses retournées
- Utiliser les mots-clés `async` et `await` pour écrire du code asynchrone plus lisible et maintenable
- Manipuler les réponses HTTP pour extraire les données retournées
- Expliquer la relation client / serveur dans le cadre de l'utilisation d'une API
- Construire des URL complètes avec les paramètres nécessaires (query parameters, path parameters).
- Gérer les réponses des API par la manipulation de JSON

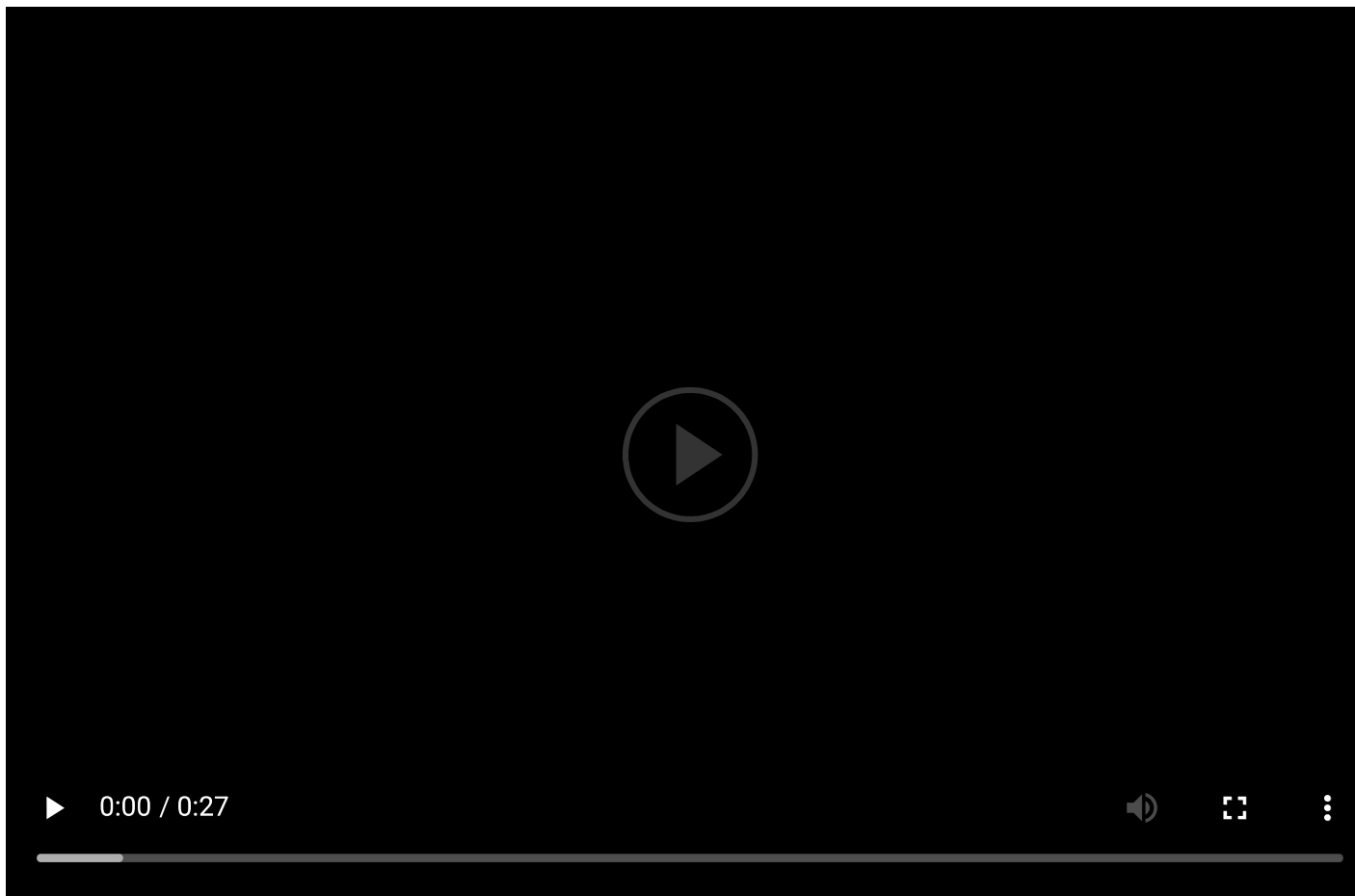
## Enoncé

Il est temps de voir ce que tu as assimilé du projet "Dataviz" 💪

### Mini-station Météo

L'objectif de ce kata est de faire une mini-station météo qui donne la température actuelle dans une ville donnée !

Voici une démonstration du résultat final que tu es censé-e obtenir à la fin des quatre étapes de ce kata.



## Etape 1: Récupère le code source de base

Tu as deux options pour récupérer les fichiers de base 📌

- Faire un fork [du projet Github](#) (et cloner le fork récupéré sur ton compte en local)
- Copier-coller les trois fichiers [de ce repository](#) sur ton ordinateur

⚠️ Quelque soit l'option choisie, tu verras que le CSS et le HTML te sont fournis.

Tu ne dois te concentrer que sur la partie Javascript pour ce kata !

## Etape 2: Récupérer les coordonnées de la ville

En tant qu'utilisateur :

- Je saisis une ville dans le champ texte
- Je clique sur le bouton "OK"

Je vois alors s'afficher sur la page les coordonnées GPS de la ville



L'utilisateur a saisi la ville d'Agadir, on voit les coordonnées GPS de la ville

Tu vas pouvoir récupérer ces informations avec l'**API de Geocoding de Open Street Map**. Pour une ville donnée, elle te retournera différentes informations dont les coordonnées GPS (propriétés **lat** et **lon**).

La documentation de cette API est fournie [ici](#).

### **i** Exemples d'appels pour les villes de Paris, Nantes et Lyon :

- <https://nominatim.openstreetmap.org/search?q=Paris&format=json&addressdetails=1&limit=1>
- <https://nominatim.openstreetmap.org/search?q=Lyon&format=json&addressdetails=1&limit=1>
- <https://nominatim.openstreetmap.org/search?q=Nantes&format=json&addressdetails=1&limit=1>

## Etape 3: Affiche la température courante de la ville

Maintenant qu'on a réussi à afficher les coordonnées, il est temps d'afficher la température !



En tant qu'utilisateur·ice:

- Je saisis une ville dans le champ texte
- Je clique sur le bouton "OK"

Je vois alors s'afficher sur la page les coordonnées GPS ET la température actuelle de la ville.

#### ⚠ Contrainte technique imposée

Les deux appels d'API doivent se faire dans deux fonctions distinctes:

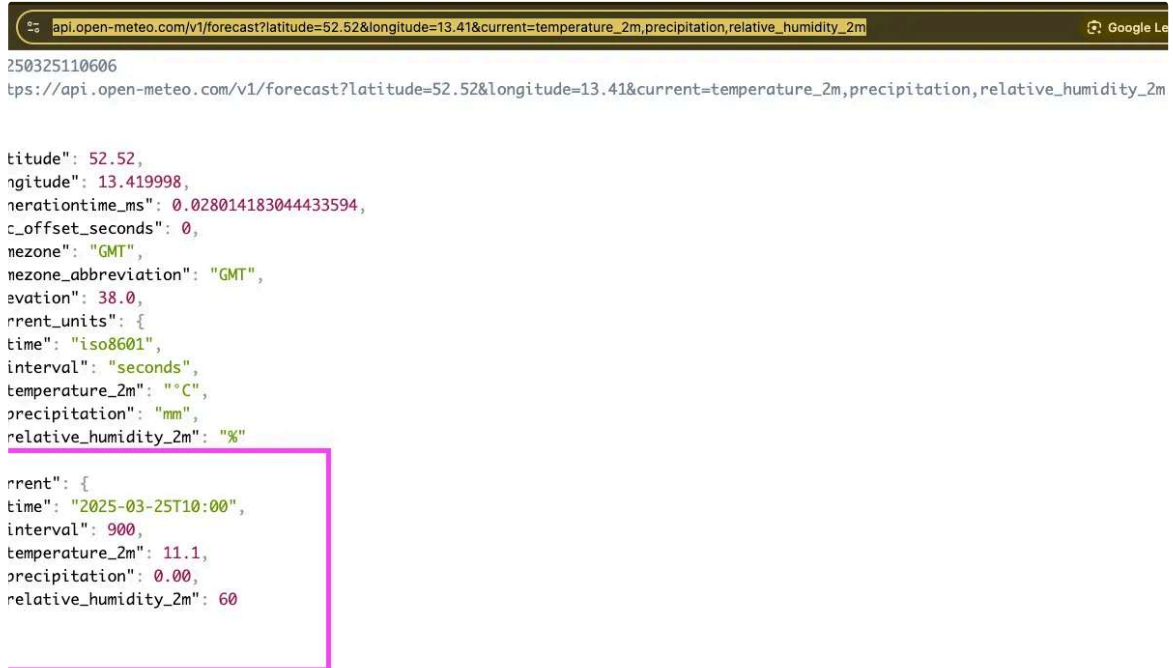
`fetchCoordinates` et `fetchWeather` !

L'API "Météo" ne peut se déclencher que si l'API "Coordonnées GPS" a bien fait son travail ...

Tu vas utiliser l'API Open-Météo pour récupérer les informations liées à la température. Elle prend en paramètre une *latitude* et une *longitude*. Tu es censé·e avoir une fonction faite à l'étape précédente qui te permet de récupérer ces infos 🧐

### Exemple d'appel pour la ville de Berlin

- [https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&current=temperature\\_2m,precipitation,relative\\_humidity\\_2m](https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&current=temperature_2m,precipitation,relative_humidity_2m)



```

250325110606
https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&current=temperature_2m,precipitation,relative_humidity_2m

{
  "latitude": 52.52,
  "longitude": 13.419998,
  "generationtime_ms": 0.028014183044433594,
  "c_offset_seconds": 0,
  "timezone": "GMT",
  "timezone_abbreviation": "GMT",
  "elevation": 38.0,
  "current_units": {
    "time": "iso8601",
    "interval": "seconds",
    "temperature_2m": "°C",
    "precipitation": "mm",
    "relative_humidity_2m": "%"
  },
  "current": {
    "time": "2025-03-25T10:00",
    "interval": 900,
    "temperature_2m": 11.1,
    "precipitation": 0.00,
    "relative_humidity_2m": 60
  }
}

```

Les informations liées à la température, la précipitation et l'humidité se trouvent dans la propriété "current"

La documentation technique d'Open-Météo se trouve [ici](#)

## Etape 3 : Gère les erreurs

En tant qu'utilisateur-ice:

- Je saisis n'importe quoi dans le champ texte (ex: "DFGoiuretkjh")
- Je clique sur le bouton "OK"

Je dois voir un message d'erreur "Ville non trouvée".

**Contrainte technique:**

L'API "Weather" ne doit pas se déclencher puisque l'API Open Street Map n'a pas retourné de coordonnées saines.



Exemple d'état d'erreur

- Tu as fini ce kata et tu souhaites aller plus loin ?