

1. **Modifiez la requête actuelle** pour interroger dynamiquement plusieurs villes successivement via le Runner de Postman. Comment automatiser cela sans modifier la requête à chaque fois ?

On peut créer un fichier .csv contenant le nom des villes à tester et le passer au runner Postman pour qu'il appelle l'endpoint pour chaque ville de la liste

```
city
Paris
London
Tokyo
New York
```

1. **Stockez dans une variable d'environnement** la description de la météo (weather[0].main). Proposez ensuite un script conditionnel qui affiche un message spécifique selon les cas (Rain, Clear, etc.).

```
const data = pm.response.json();

pm.environment.set("lastWeather", data.weather[0].main);

if(pm.environment.get("lastWeather") == "clear"){
  console.log("Le ciel est dégagé");
}
else if(pm.environment.get("lastWeather") == "rain"){
  console.log("Il pleut des cordes");
}
else if(pm.environment.get("lastWeather") == "clouds"){
  console.log("Le ciel est nuageux");
}
else{
  console.log("Le ciel est mystérieux");
}
```

1. **Exploitez la donnée feels_like** fournie dans le bloc main. Affichez dans la console la différence entre temp et feels_like et interprétez ce que cela peut signifier dans un système IA contextuel.

```
const temp_appreciation_difference = (data.main.feels_like -
data.main.temp)

if(temp_appreciation_difference < 0){
  console.log("La température ressentie est inférieure de " +
Math.abs(temp_appreciation_difference) + " degrés à la température
réelle")
}
if(temp_appreciation_difference > 0){
  console.log("La température ressentie est supérieure de " +
Math.abs(temp_appreciation_difference) + " degrés à la température
réelle")
}
```

```

}
else{
  console.log("La température ressentie est identique à la
température réelle")
}

```

Une température ressentie fortement inférieure peut signifier qu'il y a du vent, une température largement supérieure un air humide.

1. **Enregistrez dans une collection JSON structurée** les températures de 5 villes, avec leur nom, température, humidité et état du ciel. Comment pourriez-vous transformer ce JSON en dataset utilisable pour un modèle prédictif météo ?

```

let storedCollection = pm.environment.get("weather_responses");
storedCollection = storedCollection ? JSON.parse(storedCollection) :
[];

storedCollection.push({
  city: pm.variables.get("city"),
  temp: data.main.temp,
  weather: data.weather[0].main,
  humidity: data.main.humidity
});

pm.environment.set("weather_responses",
JSON.stringify(storedCollection));

```

On devrait le transformer en DataFrame

```

import json

import pandas as pd

with open('data.json', 'r', encoding='utf-8') as f:
  data = json.load(f)

df = pd.DataFrame(data)

```

1. **Ajoutez un script de vérification** qui signale si la température dépasse une valeur critique (ex. 40 °C). Si c'est le cas, générez une alerte simulée.

```

if (data.main.temp > 40) {
  pm.test("Alerte canicule !", () => {
    pm.expect(data.main.temp).to.be.above(40);
  });
  console.log("Alerte canicule !")
}

```

1. **Récupérez la météo actuelle et l'heure de lever/coucher du soleil**, puis ajoutez un test logique qui déduit s'il fait jour ou nuit au moment de l'appel API.

```
let storedCollection = pm.environment.get("weather_responses");
storedCollection = storedCollection ? JSON.parse(storedCollection) :
[];

storedCollection.push({
  city: pm.variables.get("city"),
  temp: data.main.temp,
  weather: data.weather[0].main,
  humidity: data.main.humidity,
  sunrise: data.sys.sunrise,
  sunset: data.sys.sunset
});

pm.environment.set("weather_responses",
JSON.stringify(storedCollection));

if((data.dt > data.sys.sunrise) & (data.dt < data.sys.sunset)){
  console.log("Il fait jour à " + pm.variables.get("city"))
}
else{
  console.log("Il fait nuit à " + pm.variables.get("city"))
}
```

1. **Comparez deux villes** (ex. Alger vs Montréal) sur au moins 3 indicateurs météo et générez une sortie structurée dans la console. Comment automatiser cette comparaison dans une collection ?

```
weather_description = "Le ciel est mystérieux"
if(pm.environment.get("lastWeather") == "clear"){
  weather_description = "Le ciel est dégagé"
}
else if(pm.environment.get("lastWeather") == "rain"){
  weather_description = "Il pleut des cordes"
}
else if(pm.environment.get("lastWeather") == "clouds"){
  weather_description = "Le ciel est nuageux"
}

console.log("Il fait " + data.main.temp + " degrés à " + data.name + "
tandis qu'il fait " + pm.environment.get("lastTemp") + " degrés à " +
pm.environment.get("lastCity"))
console.log("L'humidité est de " + data.main.humidity + " % à " +
data.name + " tandis qu'ele est de " +
pm.environment.get("lastHumidity") + " % à " +
pm.environment.get("lastCity"))
console.log(weather_description + " à " + data.name + " tandis que " +
pm.environment.get("lastWeatherDescription") + " à " +
pm.environment.get("lastCity"))
```

```
// ☐ Ici tu extrais la température pour la réutiliser :  
pm.environment.set("lastCity", data.name);  
pm.environment.set("lastTemp", data.main.temp);  
pm.environment.set("lastWeather", data.weather[0].main);  
pm.environment.set("lastHumidity", data.main.humidity);  
pm.environment.set("lastWeatherDescription", weather_description)
```