**1.How to compare two Json have the same properties without order?**

**a.let obj1={name: "Person1", age:5};**

**b.let obj2={age:5, name: "Person1"};**

To compare two JSON objects without order, you can use the JSON.stringify() method to convert both objects into strings and then compare the strings.

let obj1 = {name: "Person1", age: 5};

let obj2 = {age: 5, name: "Person1"};

if (JSON.stringify(obj1) === JSON.stringify(obj2)) {

console.log("The objects are equal.");

} else {

console.log("The objects are not equal.");

}

JSON.stringify() method is used to convert both obj1 and obj2 into strings. The resulting strings are then compared using the === operator. Since the order of the properties in the objects doesn't matter, the comparison will return true and the message "The objects are equal" will be logged to the console. this method assumes that the property names and values in the objects are the same. If the objects have different properties or different values for the same properties, the comparison will return false.

**2. use the rest countries API URL -> https://restcountries.com/v3.1/all and display all the country flags in the console**

fetch("https://restcountries.com/v3.1/all")

.then(response => response.json())

.then(data => {

// Loop through the array of countries

data.forEach(country => {

// Display the flag URL in the console

console.log(country.flags.svg);

});

})

.catch(error => {

console.error("Error fetching data:", error);

});

fetch() function to make a GET request to the Rest Countries API URL. Once the response is received, we use the .json() method to parse the response data into a JavaScript object. Then we use a forEach() loop to iterate through the array of countries in the response data. For each country, we access the flags.svg property to get the URL of the flag image, and we display the URL in the console using console.log(). If an error occurs during the fetch request, we log an error message to the console using console.error().

**3. use the rest countries API URL https://restcountries.com/v3.1/all and print all countries names, regions, sub-region and populations**

fetch("https://restcountries.com/v3.1/all")

.then(response => response.json())

.then(data => {

// Loop through the array of countries

data.forEach(country => {

// Print the country name, region, sub-region, and population

console.log(`Name: ${country.name.common}`);

console.log(`Region: ${country.region}`);

console.log(`Sub-region: ${country.subregion}`);

console.log(`Population: ${country.population}`);

console.log("--------------");

});

})

.catch(error => {

console.error("Error fetching data:", error);

});

fetch() function to make a GET request to the Rest Countries API URL. Once the response is received, we use the .json() method to parse the response data into a JavaScript object. Then we use a forEach() loop to iterate through the array of countries in the response data. For each country, we access the name.common, region, subregion, and population properties to get the country name, region, sub-region, and population, respectively. We then print this information to the console using console.log(). We also add a line of dashes (--------------) after each country's information to make it easier to read. If an error occurs during the fetch request, we log an error message to the console using console.error().