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

In JavaScript, you can work with JSON data using two main methods: parsing JSON data into JavaScript objects and serializing JavaScript objects into JSON format.

**1.Parsing JSON into JavaScript Objects:**

You can use the JSON.parse() method to parse a JSON string into a JavaScript object. Here's an example:

**const jsonString = '{"name": "Alice", "age": 30, "city": "New York"}';**

**const parsedData = JSON.parse(jsonString);**

**console.log(parsedData.name); // Outputs: Alice**

**console.log(parsedData.age); // Outputs: 30**

**2.Serializing JavaScript Objects to JSON:**

You can use the JSON.stringify() method to convert a JavaScript object into a JSON-formatted string. Here's an example:

**const data = {**

name**: "Bob",**

**age: 25,**

**city: "Los Angeles"**

**};**

**const jsonString = JSON.stringify(data);**

**console.log(jsonString); // Outputs: '{"name":"Bob","age":25,"city":"Los Angeles"}'**

When comparing two JSON objects in JavaScript, you can use the equality (===) operator or a deep equality function like \_.isEqual from a library like lodash to compare the **objects**. Here's an example using the equality operator:

**const json1 = '{"name": "Alice", "age": 30, "city": "New York"}';**

**const json2 = '{"age": 30, "name": "Alice", "city": "New York"}';**

**const obj1 = JSON.parse(json1);**

**const obj2 = JSON.parse(json2);**

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

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

**} else {**

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

**}**

**T**his code first parses the JSON strings into JavaScript objects and then uses JSON.stringify to convert them back to JSON strings for comparison. If the order of properties doesn't matter, this approach will consider the two JSON objects equal.

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

To fetch and display country flags from a Rest Countries API, you would need the correct API endpoint. Assuming that the API endpoint is still in use and correct, you can fetch and display country flags as follows:

**// Define the correct API URL**

**const apiUrl = 'https://restcountries.com/v3.1/all';**

**// Fetch data from the API**

**fetch(apiUrl)**

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

**.then((data) => {**

**// Iterate through the data and log the country flags**

**data.forEach((country) => {**

**const flag = country.flags.png;**

**console.log(`Country: ${country.name.common}, Flag: ${flag}`);**

**});**

**})**

**.catch((error) => {**

**console.error('Error fetching data:', error);**

**});**

Please double-check the API URL to ensure it is correct, and then run this code in a browser or a JavaScript environment that supports the Fetch API. It will fetch data from the Rest Countries API, iterate through the response data, and log the country names and flag URLs to the console.

**3.Use same rest countries and print all the name region,sub-region and population**

Certainly! To fetch and print the name, region, sub-region, and population of each country from the Rest Countries API, you can use the following JavaScript code:

**// Define the API URL**

**const apiUrl = 'https://restcountries.com/v3.1/all';**

**// Fetch data from the API**

**fetch(apiUrl)**

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

**.then((data) => {**

**// Iterate through the data and print the desired information**

**data.forEach((country) => {**

**const name = country.name.common;**

**const region = country.region;**

**const subRegion = country.subregion;**

**const population = country.population;**

**console.log(`Country: ${name}`);**

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

**console.log(`Sub-Region: ${subRegion}`);**

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

**console.log('----------------------');**

**});**

**})**

**.catch((error) => {**

**console.error('Error fetching data:', error);**

**});**

This code will fetch data from the Rest Countries API, iterate through the response data, and print the name, region, sub-region, and population of each country in the console.