1. ***Solving problems using array functions on rest countries data.***
   1. ***Get all the countries from Asia continent /region using Filter function***

***var xhr = new XMLHttpRequest();***

***xhr.onload = function(){***

***if (xhr.status >= 200 && xhr.status <= 400){***

***s var temp = JSON.parse(xhr.responseText);***

***//used filter to filter asian countries then used chaining map to only get names of countries***

***let result = temp.filter(x=> x.region === "Asia").map(x => {return x.name} )***

***//printing the result***

***console.log(result);***

***}else{***

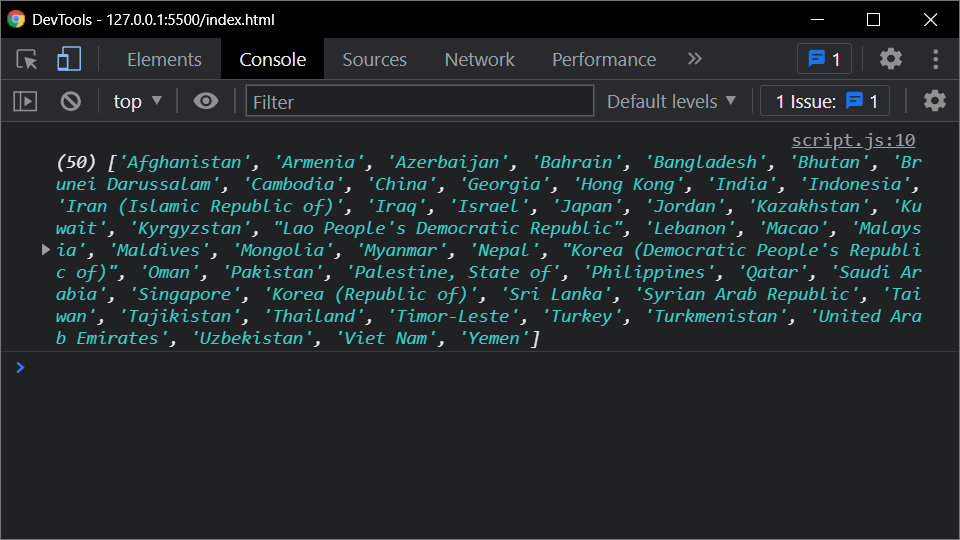
***console.log(xhr.responseText);***

***}***

***};***

***xhr.open("GET","https://restcountries.eu/rest/v2/all")***

***xhr.send()***

******

* 1. ***Get all the countries with a population of less than 2 lakhs using Filter function***

***var xhr = new XMLHttpRequest();***

***xhr.onload = function(){***

***if (xhr.status >= 200 && xhr.status <= 400){***

***var temp = JSON.parse(xhr.responseText);***

***/\*used filter to filter countries having population less than 2 lakh***

***then used chaining map to only get names of countries\*/***

***let result = temp.filter(x=> x.population < 200000).map(x => {return x.name} )***

***//printing the result***

***console.log(result);***

***}else{***

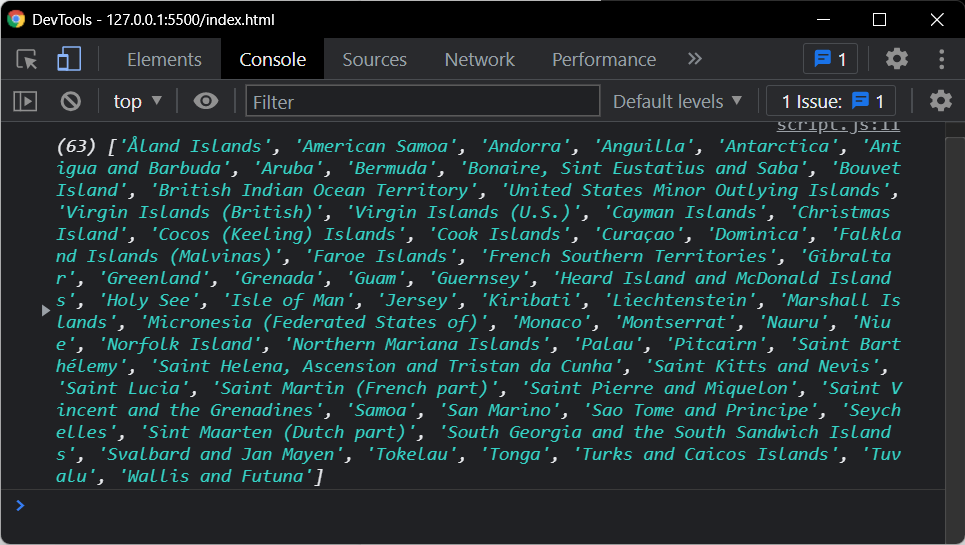
***console.log(xhr.responseText);***

***}***

***};***

***xhr.open("GET","https://restcountries.eu/rest/v2/all")***

***xhr.send()***

******

* 1. ***Print the following details name, capital, flag using forEach function***

***var xhr = new XMLHttpRequest();***

***xhr.onload = function(){***

***if (xhr.status >= 200 && xhr.status <= 400){***

***var temp = JSON.parse(xhr.responseText);***

***//creating an array for storing result***

***let result = []***

***//using foreach for iteration***

***temp.forEach((x) => {***

***//pushing name,capital and flag into result array using call back function***

***result.push({"name": x.name, "capital":x.capital, "flag":x.flag})***

***})***

***//printing the result***

***console.log(result)***

***}else{***

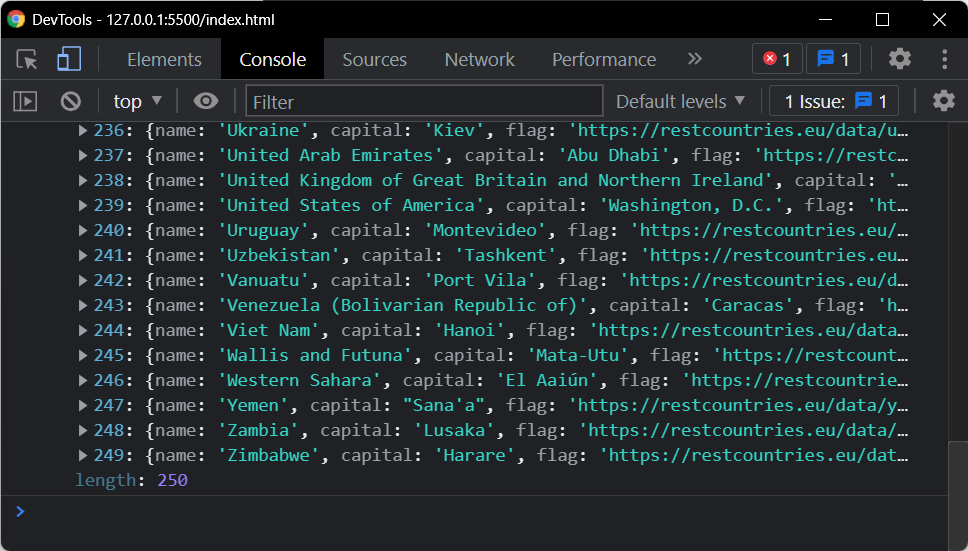
***console.log(xhr.responseText);***

***}***

***};***

***xhr.open("GET","https://restcountries.eu/rest/v2/all")***

***xhr.send()***

******

* 1. ***Print the total population of countries using reduce function***

***var xhr = new XMLHttpRequest();***

***xhr.onload = function(){***

***if (xhr.status >= 200 && xhr.status <= 400){***

***var temp = JSON.parse(xhr.responseText);***

***/\*used map to get population of all the countries***

***then used chaining reduce to make total of it\*/***

***let result = temp.map(x => {return x.population}).reduce((a,b) => a + b)***

***//printing the result***

***console.log(result);***

***}else{***

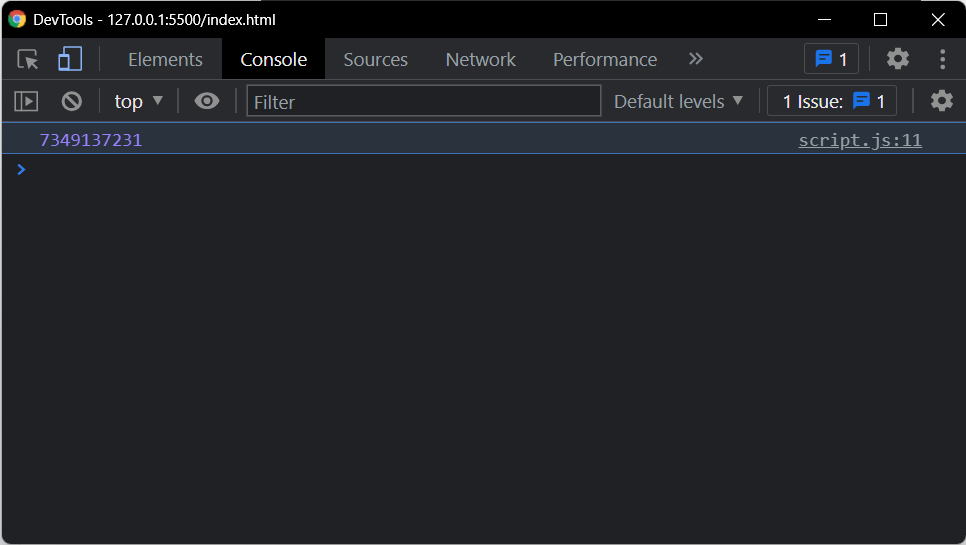
***console.log(xhr.responseText);***

***}***

***};***

***xhr.open("GET","https://restcountries.eu/rest/v2/all")***

***xhr.send()***

******

* 1. ***Print the country which uses US Dollars as currency***

***var xhr = new XMLHttpRequest();***

***xhr.onload = function(){***

***if (xhr.status >= 200 && xhr.status <= 400){***

***var temp = JSON.parse(xhr.responseText);***

***//filtering to get countries***

***let result = temp.filter(x => {***

***let curr = x.currencies //grtting vurrencies object into variable***

***let flag = false //setting flag to false as default***

***//maping to iterate***

***curr.map(y => {***

***//checking of currencies***

***if(y.name === "United States dollar" || y.name === "United State Dollar")***

***{***

***flag = true***

***}***

***})***

***return flag***

***}).map(x => {return x.name} ) //maping to get only names***

***//printing the result***

***console.log(result)***

***}else{***

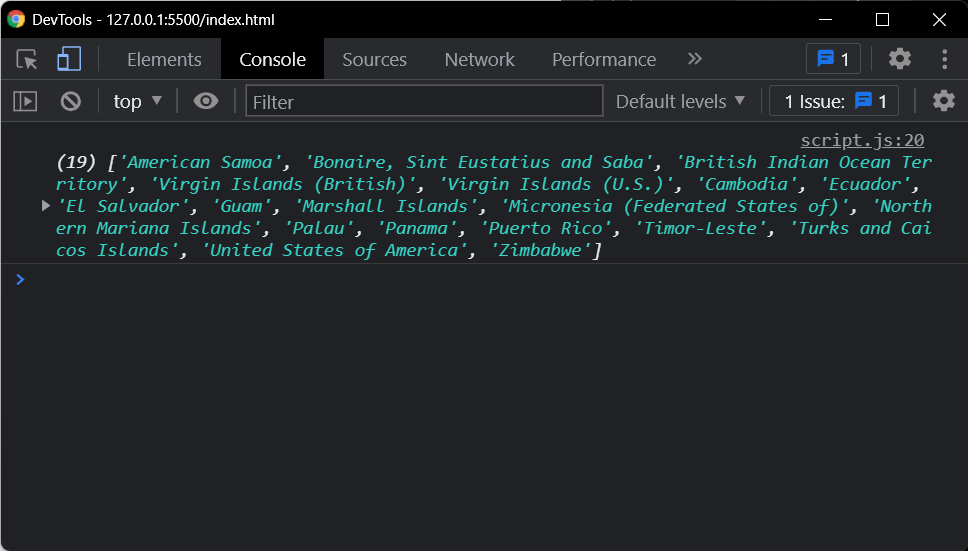
***console.log(xhr.responseText);***

***}***

***};***

***xhr.open("GET","https://restcountries.eu/rest/v2/all")***

***xhr.send()***

******