TASK2

1).html and script.js file and run a for loop on the data and print all the country names in the console.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script src="script.js"></script>
</body>
</html>
```

Script.js

```
//create a request variable
var request =new XMLHttpRequest();
//connection
request.open('GET','https://restcountries.eu/rest/v2/all','TRUE');
//send a request
request.send();
//load data
request.onload=function(){
    var data=JSON.parse(this.response);
        for(var i in data)
        console.log(data[i].name);
}
```

2). Write a write up on Difference between copy by value and copy by reference.
Consider two data types, primitive and composite.
a. Primitive data types are passed by value . For example, consider the below code
var a = 5;
var b = a;
console.log(a,b) \\ 5, 5
Here when ‘a’ is assigned to b, it is copied just by value meaning if you change b again, it doesn’t necessarily change the value of a. See the code below
var a = 5;
var b = a;
b = 10;

console.log(a,b) \\ 5, 10
We can clearly see the difference from above.
b. Composite data are generally passed by reference ie when we create an array as below
var array = [1];
var temp = array;
console.log(array, temp) \\ [1], [1]
Here while we create an array, it is assigned an address of the location in memory. So when we assign array to temp, temp is copied by reference to the address. Now if we manipulate temp, both temp and arr are changed because they are both pointing to the same reference.
var array = [1];
var temp = array;
temp.push(2);
console.log(array, temp) \\ [1,2], [1,2]

3). How to copy by value a composite data type (array+objects).

a. Using the spread operator
var array = [1];
var temp = [...array];
temp[0] = 2;
console.log(array, temp) \\ [1], [2]

b. Using Object.assign()
var array = [1];
var temp = Object.assign([], array);
temp[0] = 2;
console.log(array, temp) \\ [1], [2]
c. Using JSON.parse() and JSON.stringify();

var array = [1];
var temp = JSON.parse(JSON.stringify(array));
temp[0] = 2;
console.log(array, temp) \\ [1], [2]

4).JSON task
https://medium.com/@reach2arunprakash/guvi-zen-code-sprint-javascript-practice-problems-in-json-objects-and-list-49ac3356a8a5
Problem-0 PART-A
Script.js

```
var cat = {
    name: 'Fluffy',
    activities: ['play', 'eat cat food'],
```

```
    catFriends: [
    {
    name: 'bar',
    activities: ['be grumpy', 'eat bread omblet'],
    weight: 8,
    furcolor: 'white'
    },
    {
    name: 'foo',
    activities: ['sleep', 'pre-sleep naps'],
    weight: 3
    }
    ]
   }
//height and weight
cat.height='2 feet';
    cat.weight= 10;
    //name updation

   cat.name='Fluffyy';
   console.log(cat);
   //activities of fluffy's friends
       for(var j=0;j<cat.catFriends.length;j++)
   console.log(cat.catFriends[j].activities);
//catfriend names

  for(var j=0;j<cat.catFriends.length;j++)
  console.log(cat.catFriends[j].name);
   //weight of catfriends
  var sum=0;
  for(var j=0;j<cat.catFriends.length;j++)
  sum+=cat.catFriends[j].weight;
  console.log(sum);

//total activities
console.log(cat.activities.length+cat.catFriends[0].activities.length+cat.
catFriends[1].activities.length);
//add activities
      cat.catFriends[0].activities.push('Drooling');
 cat.catFriends[0].activities.push('Playing');
```

```
 cat.catFriends[1].activities.push('Running');
 cat.catFriends[1].activities.push('Eating pedigree');
 for(var j=0;j<cat.catFriends.length;j++)
 console.log(cat.catFriends[j].activities);
//change fur colour
cat.catFriends[0].furcolor='black';
console.log(cat.catFriends[0].furcolor);
```

PART B

Script.js

```
var myCar = {
    make: 'Bugatti',
    model: 'Bugatti La Voiture Noire',
    year: 2019,
    accidents: [
    {
    date: '3/15/2019',
    damage_points: '5000',
    atFaultForAccident: true
    },
    {
    date: '7/4/2022',
    damage_points: '2200',
    atFaultForAccident: true
    },
    {
    date: '6/22/2021',
    damage_points: '7900',
    atFaultForAccident: true
    }
    ]
   }
   for(var i=0;i<myCar.accidents.length;i++){
       myCar.accidents[i].atFaultForAccident=false;
       console.log(myCar.accidents[i].atFaultForAccident);
       }

       for(var i=0;i<myCar.accidents.length;i++)
       console.log(myCar.accidents[i].date);
```

PROBLEM 2

```
var obj=[{name: 'RajiniKanth', age: 33, hasPets : false},
{name:'NTR', age: 30, hasPets : true}];
function printing(obj){
    for(var i=0;i<obj.length;i++)
    console.log(obj[i]);
}
printing(obj);
function keys(obj){
   for(var i in obj)
console.log(Object.keys(obj[i]));

}
keys(obj);
```

PROBLEM 3

```
var obj = {name: 'ISRO', age: 35, role: 'Scientist'};
function convertListToObject(obj) {
    let list = [];
    for(let keys in obj){
        list.push([keys, obj[keys]]);
    }
    return list;
}

console.log(convertListToObject(obj));
```

PROBLEM 4

```
var arr = ['GUVI', 'I', 'am', 'a geek'];
function transformFirstAndLast(arr) {
    let newObject={};
    let p=arr.length-1;
    newObject[arr[0]]=arr[p];
 return newObject;
}
console.log(transformFirstAndLast(arr));
```

PROBLEM 5

```
var arr = [['make', 'Ford'], ['model', 'Mustang'], ['year', 1964]];
```

```
function fromListToObject(arr) {
 var newObject = {};
 for(let i in arr){
   //  newObject[arr[i]]=Object.values(arr[i]);
   newObject[arr[i][0]] =  arr[i][1];
 }
 return newObject;
}
console.log(fromListToObject(arr));
```

PROBLEM 6

```
var arr= [[['firstName', 'Vasanth'], ['lastName', 'Raja'], ['age', 24],
['role', 'JSWizard']], [['firstName', 'Sri'], ['lastName', 'Devi'],
['age', 28], ['role', 'Coder']]];
function transformEmployeeData(arr) {
 var transformEmployeeList = [];
 for(let i=0;i<arr.length;i++){
      transformEmployeeList[i]={};
     for(let j=0;j<arr[i].length;j++)
     transformEmployeeList[i][arr[i][j][0]]=arr[i][j][1];
 }
 //Your code

 return transformEmployeeList;
}
console.log(transformEmployeeData(arr));
```

PROBLEM 7

```
var expected = {foo: 5, bar: 6};
var actual = {foo: 5, bar: 6};
function assertObjectsEqual(actual, expected, testName){
    let actualstring =JSON.stringify(actual);
    let expectedstring =JSON.stringify(expected);
    if(actualstring!==expectedstring)
    console.log('FAILED[' + testName +'] Expected ' + expectedstring +
',but got '+actualstring);
    else
    console.log('Passed');
}
assertObjectsEqual(actual,expected,'detects that two objects are equal');
```

PROBLEM 8

```
var securityQuestions = [
    {
    question: 'What was your first pet's name?',
    expectedAnswer: 'FlufferNutter'
    },
    {
    question: 'What was the model year of your first car?',
    expectedAnswer: '1985'
    },
    {
    question: 'What city were you born in?',
    expectedAnswer: 'NYC'
    }
   ]
   function chksecurityQuestions(securityQuestions,question,ans) {
   for(let i in securityQuestions){
       if(securityQuestions[i].question===question){
           if(securityQuestions[i].expectedAnswer===ans)
             return true;
       }
   }
   return false;
    }
   //Test case1:
   var ques = 'What was your first pet's name?';
   var ans  =  'FlufferNutter';
   var status = chksecurityQuestions(securityQuestions, ques, ans);
   console.log(status); // true
   //Test case2:
   var ques = 'What was your first pet's name?';
   var ans  =  'DufferNutter';
   var status = chksecurityQuestions(securityQuestions, ques, ans);
   console.log(status); // flase
```

PROBLEM 9

```
var students = [
    {
    name: 'Siddharth Abhimanyu', age: 21}, { name: 'Malar', age: 25},
```

```
    {name: 'Maari',age: 18},{name: 'Bhallala Deva',age: 17},
    {name: 'Baahubali',age: 16},{name: 'AAK chandran',age: 23},
{name:'Gabbar Singh',age: 33},{name: 'Mogambo',age: 53},
    {name: 'Munnabhai',age: 40},{name: 'Sher Khan',age: 20},
    {name: 'Chulbul Pandey',age: 19},{name: 'Anthony',age: 28},
    {name: 'Devdas',age: 56}
    ];
   function returnMinors(students)
   {
       let a=[];
       for(let i in students){
           if(students[i].age<20)
           a.push(students[i].name);

       }
       return a;
   }
   console.log(returnMinors(students));
```

5) Try the rest countries api. Extract and print the total population of all the countries in the console. use the html template. https://restcountries.eu/rest/v2/all

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script src="script.js"></script>
</body>
</html>
```

script.js

```
var request =new XMLHttpRequest();
//connection
request.open('GET','https://restcountries.eu/rest/v2/all','TRUE');
//send a request
```

```
request.send();
//load data
request.onload=function(){
    var data=JSON.parse(this.response);
    console.log(data);
       var sum=0;
        for(var i in data){
            sum+=data[i].population;
            console.log(data[i].population);
        }
        console.log(sum);

}
```