Task2

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

Ans :

INDEX.html :-

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>sample api data</title>
</head>
<body>
   <script src ="script.js">?</script>
</body>
</html
```

SCRIPT.js :-

```
var request = new XMLHttpRequest();

request.open('GET', 'https://restcountries.eu/rest/v2/all',true);

request.send();

request.onload = function(){
    var data = JSON.parse(this.response);
    console.log(data);
    for(let i in data)
    {
    console.log(data[i].name);
    }
    }
```

Que.2 : Write a write up on Difference between copy by value and copy by reference.

Ans :

Copy by value

- In a primitive data-type when a variable is assigned a value we can imagine that a box is created in the memory. This box has a sticker attached to it i.e. the variable name. Inside the box the value assigned to the variable is stored.

```
 For example:  var x= 10;
                         var y='amar';
                         console.log(x,y);
                          var a=x;
                           var x=3;
                          console.log(a,x,y);
->10 'amar'
->10 3 'amar'
```

- In above example, In first 2 lines ,x' contains value 10, 'y' contains 'amar'. Then in the fourth line the **value** in the box 'x' is copied into the variable 'a'.
- At this point of time both 'x' and 'a' contains the value 10. However, an important thing to understand here is that even though 'x' and 'a' contains the same value they are not connected to each other. It is so because the values are directly copied into the new variables.
- Changes taking palace in one does not affect the other.

Copy by reference

- In case of a non-primitive data-type the values are not directly copied. When a non-primitive data-type is assigned a value a box is created with a sticker of the name of the data-type. However, the values it is assigned is not stored directly in the box. The language itself assigns a different memory location to store the data. The address of this memory location is stored in the box created.

```
For example: let user={name : 'person1'};
                   let admin= user;
                   admin.name = 'person2';
                    console.log(user.name);
-> personal
```

- In the above example, when the value of admin is changed it automatically changes the value of user as well.
- This happens because both 'user' and 'admin' are storing the address of the memory location. And when one changes the values in the allocated memory it is reflected in the other as well.
- We can further elaborate it we can say that; copy by reference is like having two keys of the same room shared between 'admin' and 'user'.   one of them alters the arrangement of the room the other would experience it as well.

Que.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.

Ans :

INDEX.html :-

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>sample api data</title>
</head>
<body>
    <script src ="script.js">?</script>
</body>
</html
```

SCRIPT.js :-

```
var request = new XMLHttpRequest();

request.open('GET', 'https://restcountries.eu/rest/v2/all',true);

request.send();

request.onload = function(){
    var data = JSON.parse(this.response);
    console.log(data);
    for(let i in data)
    {
    console.log(data[i].population);
    }
    }
```

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

Ans :

- The **Object.assign()** method copies all enumerable own properties from one or more source objects to a target object. It returns the target object. Note this will be a shallow copy.

For example:
```
var arr=[1,2,3];
var b= Object.assign([],arr);
console.log(arr,b);
 b[2]=100;
console.log(arr,b);
```
```
-> [1,2,3] [1,2,3]
-> [1,2,3] [1,2,100]
```

- Note the empty [] as the first argument, this will ensure you don't mutate the original object

Que.4 : JSON task https://medium.com/@reach2arunprakash/guvi-zen-code-sprint-javascript-practice-problems-in-json-objects-and-list-49ac3356a8a5

Ans :

- var cat = {

```
    name: 'Fluffy',
    activities: ['play','eat cat food'],
    catFriends: [{
        name: 'bar',
        activities: ['be grumpy','eat bread omlet'],
        weight: 8,
        furcolor: 'white'
    },
    {
        name: 'foo',
        activities: ['sleep', 'pre-sleep naps'],
        weight:3
    }
    ]
};
console.log(cat);
cat.height=5.8;
cat.weight=9;
cat.name='fluffyy';
for(let i in cat.catFriends){
    console.log(cat.catFriends[i].activities);
     console.log(cat.catFriends[i].name);
}

for(let i in cat){
    console.log(cat.activities);
     for(let j in cat.catFriends){
         console.log(cat.catFriends[j].activities);
     }
}
for(let i in cat.catFriends){
    cat.catFriends[0].furcolor='black';

}
console.log(cat.catFriends);

 let totalWeight = 0;
for(let i in cat.catFriends){

   totalWeight = cat.catFriends[0].weight + cat.catFriends[1].weight;
}
console.log(totalWeight);
```

- 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(let i in myCar.accidents){
   myCar.accidents[i].atFaultForAccident=false;
   console.log(myCar.accidents[i].date);
}
console.log(myCar);
1.  var object = {name: "RajiniKanth", age: 33, hasPets : false};
    function printAllValues(obj) {
     console.log(Object.values(object));
    }
    printAllValues();
2.  var object ={name : 'RajiniKanth', age : 25, hasPets : true};
    function printAllKeys(obj) {
     console.log(Object.keys(object));
    }
    printAllKeys();
3.  var object = {name: 'ISRO', age: 35, role: 'Scientist'};
    function convertListToObject(obj) {

     console.log(Object.entries(object));
    }
    convertListToObject();
4.  var array = ['GUVI', 'I', 'am', 'a geek'];
    function transformFirstAndLast(arr) {
```

4.
```
   let newObject = {};
   let arrLength = arr.length;
   newObject[arr[0]] = arr[arrLength-1];
   return newObject;
}
console.log(transformFirstAndLast(array));
```
5.
```
var arr = [["make", "Ford"], ["model", "mustang"], ["year", 1964]];
function fromListToObject() {
   let newObject = {};
   for(let i in arr){
      var key = arr[i][0];
      newObject[key] = arr[i][1];
   }
   console.log(newObject);
}
fromListToObject();
->{ make: 'Ford', model: 'mustang', year: 1964 }
```
6.
```
var arr= [[['firstName', 'Vasanth'], ['lastName', 'Raja'], ['age', 24], ['role', 'JSWizard']], [['firstName',
'Sri'], ['lastName', 'Devi'], ['age', 28], ['role', 'Coder']]];
function transformEmployeeData(arr) {
 var tranformEmployeeList = [];
 for(let i=0;i<arr.length;i++){
    tranformEmployeeList[i]={};
    for(let j=0;j<arr[i].length;j++)
    tranformEmployeeList[i][arr[i][j][0]]=arr[i][j][1];
 }

 return tranformEmployeeList;
}
console.log(transformEmployeeData(arr));
```
7.
```
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'
 }
]
```

7.

```
function chksecurityQuestions(securityQuestions,question,ans) {
 let output=false;
  for(let i=0;i<securityQuestions.length;i++){
      if(securityQuestions[i].question==question){
          if(securityQuestions[i].expectedAnswer==ans)
          output=true;
      }
  }
 return output;
}
var ques = 'What was your first pet’s name?';
var ans  =  'FlufferNutter';
 console.log(chksecurityQuestions(securityQuestions, ques, ans));

var ques1 = 'What was your first pet’s name?';
var ans1 =  'DufferNutter';
console.log(chksecurityQuestions(securityQuestions, ques1, ans1));
```

8.

```
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(arr)
{
    for(let i in students){
        if(students[i].age<20)
        console.log(students[i]);
    }
}
returnMinors(students);
```