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

```
var data = JSON.parse(this.response);
    //console.log(data[0].capital);
    for(i in data)
    {
        console.log(data[i].flag);
    }
    for(j in data)
    {
        if(data[j].region==='Asia')
        console.log(data[j].countries);
    }
```

2. write up on Difference between copy by value and copy by reference.
   Call-by-value:
   - ❖ This method copies a value to a local variable.
   - ❖ Change in values does not affect the values in another variable.
   - ❖ It takes seperate locations in memory.

   Call-by-reference:
   - In call by reference all objects are called by reference.
   - If a variable is made equal to another variable both share common memory locations.
   - If change in one variable will affect the other variable.
3. 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.

```
for (i in a)
    console.log(a[i].population);
```

4. How to copy by value a composite data type (array+objects).
   - Use the spread (...) syntax
   - Use the Object.assign() method
   - Use the JSON.stringify() and JSON.parse() methods
5. JSON Task
   - ➢ Problem 1: Write a function called “printAllValues” which returns an newArray of all the input object’s values.
     var object = {name: 'RajiniKanth', age: 33, hasPets : false};
     console.log(Object.values(object));

- Problem 2: Write a function called “printAllKeys” which returns an newArray of all the input object’s keys.

```
var object = {name: 'RajiniKanth', age: 33, hasPets : false};
console.log(Object.keys(object));
```

- Problem 3: Write a function called “convertObjectToList” which converts an object literal into an array of arrays.

```
var object = {name: 'ISRO', age: 35, role: 'Scientist'};
Object.entries(object);
```

- Problem 4: Write a function ‘transformFirstAndLast’ that takes in an array, and returns an object with:
  1) the first element of the array as the object’s key, and
  2) the last element of the array as that key’s value.

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

- Problem 5: Write a function “fromListToObject” which takes in an array of arrays, and returns an object with each pair of elements in the array as a key-value pair.

```
var arr = [['make', 'Ford'], ['model', 'Mustang'], ['year', 1964]];
var obj = {};
var len = arr.length;
for(var i=0;i<len;i++)
{
        j=0;
        obj[arr[i][j]] = arr[i][j+1];
}
console.log(obj);
```

- Problem 6: Write a function called “transformGeekData” that transforms some set of data from one format to another.

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

- Problem 7: Write an “assertObjectsEqual” function from scratch.
  Assume that the objects in question contain only scalar values (i.e., simple values like strings or numbers).
  It is OK to use JSON.stringify().

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

- Problem 8: I have a mock data of security Questions and Answers. You function should take the object and a pair of strings and should return if the quest is present and if its valid answer

```
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: Write a function to return the list of characters below 20 age.

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