Task-3:

Question2: Try the rest countries api. Extract and print the flag url of all the countries in the console. use the html template. https://restcountries.eu/rest/v2/all

```
<!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>
```

```
var request=new XMLHttpRequest();
request.open('GET', 'https://restcountries.eu/rest/v2/all', true)
request.send();
request.onload = function(){
    var countrydata = JSON.parse(this.response);
    var flagurl=countrydata.map((x)=>
        x.flag
 )
 console.log(flagurl);
}
```

Question1: JSON task https://medium.com/@reach2arunprakash/guvi-zen-code-sprint-javascript-practice-problems-in-json-objects-and-list-49ac3356a8a5

Basic Tasks to play with JSON

1. **Add height and weight to Fluffy**

   cat['height']=1.5;

   cat['weight']=4;

2. **Fluffy name is spelled wrongly. Update it to Fluffyy**

   cat[name]='Fluffyy';

3. **List all the activities of Fluffyy's catFriends.**

   var all=cat.catFriends[0].activities.concat(cat.catFriends[1].activities);

4. **Print the catFriends names.**

   console.log(cat.catFriends[0].name+'&'+ cat.catFriends[1].name);

5. **Print the total weight of catFriends**

```
var a=+cat.catFriends[0].weight;

var b=+cat.catFriends[1].weight;

console.log(a+b);
```

6. **Print the total activities of all cats (op:6)**

```
var all=cat.catFriends[0].activities.concat(cat.catFriends[1].activities);

var total=all.concat(cat.activities);
```

7. **Add 2 more activities to bar & foo cats**

```
cat.catFriends[0].activities.push('eating','sleeping');

cat.catFriends[1].activities.push('stealing','escaping');
```

8. **Update the fur color of bar**

```
cat.catFriends[0].furcolor='pink';
```

Problem 0 : Part B (15 mins):

Iterating with JSON object's Values

1.Loop over the accidents array. Change atFaultForAccident from true to false.

```
for(var i=0;i<myCar.accidents.length;i++){

myCar.accidents[i].atFaultForAccident='false';}
```

2. Print the dated of my accidents

```
var arr=[];
for(var i=0;i<myCar.accidents.length;i++){

arr.push(myCar.accidents[i].date);}

console.log(arr.join(" "));
```

Problem 1

Parsing an JSON object's values:

Write a function called "printAllValues" which returns an newArray of all the input object's values.

```
function printAllValues(){
console.log(Object.values(obj));
}printAllValues();
```

Problem 2(5 mins) :

Parsing an JSON object's Keys:

Write a function called "printAllKeys" which returns an newArray of all the input object's keys.

```
function printAllKeys(){
console.log(Object.keys(obj));
}printAllKeys();
```

Problem 3(7–9 mins):

Parsing an JSON object and convert it to a list:

Write a function called "convertObjectToList" which converts an object literal into an array of arrays.

```
var object = {name: 'ISRO', age: 35, role: 'Scientist'};
var arr=[];
function convertObjectToList(){
   var a=Object.keys(object);
   var b=Object.values(object);
   for(var i=0;i<a.length;i++){
      var c=a[i]+' '+b[i];
      var c=c.split(" ")
      arr.push(c);
      if(i==a.length-1)
      console.log(arr);
   }}convertObjectToList();
```

Problem 4:Parsing a list and transform the first and last elements of it:

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 array = ['GUVI', 'I', 'am', 'Geek'];

function transformFirstAndLast(){
   var a=array[0];
   var b=array[array.length-1];
   var object={};
   object[a]=b;
   console.log(object);
}transformFirstAndLast();
```

Problem 5 (7 -9 mins):

Parsing a list of lists and convert into a JSON object:

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.
Input (Array):

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

function fromListToObject(){
   var object={};
   for(var i=0;i<array.length;i++){
      object[array[i][0]]=array[i][1];
      if(i==array.length-1)
      console.log(object);
   }
}fromListToObject();
```

Problem 6 (10 mins):

Parsing a list of lists and convert into a JSON object:

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

```
var array = [[['firstName', 'Vasanth'], ['lastName', 'Raja'], ['age', 24], ['role', 'JSWizard']], [['firstName', 'Sri'], ['lastName', 'Devi'], ['age', 28], ['role', 'Coder']]];

function transformGeekData(){
   var arr=[];
   var object={};
   for(var i=0;i<array[0].length;i++){
      object[array[0][i][0]]=array[0][i][1];
      if(i==array[0].length-1){
      arr.push(object);
      object={};}}

   for(var i=0;i<array[1].length;i++){
      object[array[1][i][0]]=array[1][i][1];
      if(i==array[0].length-1){
      arr.push(object);
      object={};}}

      console.log(arr);

}transformGeekData();
```

Problem 9(20 mins): Parsing JSON objects and Compare:

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)
{
  var newObj = [];
  for (var i = 0; i< arr.length; i++){
    if (arr[i].age < 20){
      newObj.push(arr[i]);
    }
  }
  return newObj;
}
console.log(returnMinors(students));
```

Problem 7 (10 — 20 mins):

Parsing two JSON objects and Compare:

Read this : https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify

Write an "assertObjectsEqual" function from scratch.
Assume that the objects in question contain only scalar values (i.e., simple values like strings or numbers).

```
var expected = {foo: 5, bar: 6};
   var expected1 = {foo: 6, bar: 5};
   var actual = {foo: 5, bar: 6};

   function assertObjectsEqual(actual, expected, testName){
     actualStr = JSON.stringify(actual)
     expectedStr = JSON.stringify(expected)
     if(actualStr == expectedStr){
```

```
    return "Passed"
  } else{
    return "FAILED ["+testName+"] Expected "+actualStr+", but got "+expectedStr
  }
}
console.log(assertObjectsEqual(actual, expected, 'test1'))
console.log(assertObjectsEqual(actual, expected1, 'test2'))
```

Problem 8(10 mins):

Parsing JSON objects and Compare:

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, answer) {

     for (var i = 0; i < securityQuestions.length; i++)

     {

       for (keys in securityQuestions[i]){

         if(keys == "question"){

           if(securityQuestions[i].question == question && securityQuestions[i].expectedAnswer
== answer){

               return true;

           }

         }

       }

     }


   return false;

 }
```
