ZEN CLASS ASSIGNMENT 2

1. Print all the country names in the console.

   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("COUNTRIES NAMES:");
    for(var i in data){
        console.log(data[i].name);
    }

}
```

2. Difference between copy by value and copy by reference.

   2.1 copy by value:

   - It allocates separate memory location for the new object and then assigns the copied members to the new object.
   - In this way, both the objects are independent of each other and in case of any modification to either one the other is not affected.
   - If one of the objects is deleted the other still remains in the memory.

   2.2 copy by reference:

   - Reference variable is copied into a new reference variable using the assignment operator.
   - The address stored in the old reference variable is copied into the new one. This means both the old and new reference variable point to the same object in memory.
   - If the state of the object changes through any of the reference variables it is reflected for both.

3. Extract and print the total population of all the countries in the console.

   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);
   var total=0;
   console.log("COUNTRIES POPULATION:");
    for(var i in data){
        total=total+data[i].population;
        console.log("Country : "+data[i].name+" population:
"+data[i].population);
    }
   console.log("Total Population: "+total);
}
```

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

    - Using the spread(...reference) operator.
    - Using the Object.assign() method.
    - Using the JSON.stringify() and JSON.parse() methods.

5. Write a code to get the below details of Fluffyy so that I can take him to vet.

    https://github.com/viszhnu/Fluffyy-cat

6. Iterating with JSON object's Values

    https://github.com/viszhnu/car_Accident

7. Parsing an JSON object's Values

```
var obj = {name : 'RajiniKanth', age : 33, hasPets : false};
function printAllValues(obj) {

    var arr=[];

    for(var i in obj){
        arr.push(obj[i]);
    }

    return arr;
}

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

8. Parsing an JSON object's Keys:

```
function printAllValues(obj) {
    var arr=[];
    for(var i in obj){
        arr.push(i);
    }
      return arr;
}
```

9. Parsing an JSON object and convert it to a list:

```
var obj = {name: 'ISRO', age: 35, role: 'Scientist'};
function convertListToObject(obj) {

     var arr=new Array();

     for(var i in obj){

        var arr1=new Array();
        var str=""+i;
        var str1=""+obj[i];
        arr1.push(str);
        arr1.push(str1);
        arr.push(arr1);

     }
     return arr;
}

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

10. Parsing a list and transform the first and last elements of it:

```
    var arr = ['GUVI', 'I', 'am', 'a geek'];

function transformFirstAndLast(arr) {

    var obj={};
    obj[arr[0]]=arr[arr.length-1]
    return obj;

}

console.log(transformFirstAndLast(arr));
```

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

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

function fromListToObject(arr) {
 var newObject = {};
 for(var i in arr){
         newObject[arr[i][0]]=arr[i][1];
 }
 return newObject;
}

 console.log(fromListToObject(arr));
```

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

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

function transformEmployeeData(arr) {
    var tranformEmployeeList = [];

    for(var i in arr){
        var obj={};
        for(var j in arr[i]){
          obj[arr[i][j][0]]=arr[i][j][1];
        }
        tranformEmployeeList.push(obj);
    }

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

13. Parsing two JSON objects and Compare:

```
var expected = {foo: 5, bar: 6};
var actual = {foo: 5, bar: 5}
function assertObjectsEqual(actual, expected, testName){

    if(JSON.stringify(actual)===JSON.stringify(expected)){
        console.log("Passed");
    }else{
        console.log("FAILED [my test] Expected "+
        JSON.stringify(expected)+", but got "+
        JSON.stringify(actual));
    }


}

assertObjectsEqual(actual, expected,"test");
```

14. Parsing JSON objects and Compare:

```
function chksecurityQuestions(securityQuestions,question,answer)
{

    for(var i in securityQuestions){

       if(securityQuestions[i].question===question
           &&securityQuestions[i].expectedAnswer===answer){
           return true;
       }

    }

    return false;
}
```

15. Parsing JSON objects and Compare:

```
function returnMinors(arr)
  {
      var arr1=[];
      for(var i in arr){
          if(arr[i].age<20){
              arr1.push(arr[i].name)
          }
      }
     return arr1;
  }
```