JavaScript Practice problems in JSON(Objects) and List.

Reference; *https://medium.com/@reach2arunprakash/guvi-zen-code-sprint-javascript-practice-problems-in-json-objects-and-list-49ac3356a8a5*

Q1. Add height and weight to Fluffy ?

```
var cat = {
name: "Fluffy",
height: 50,
weight: 30,
activities: ["play" , "eat cat food"],
catFriends: [
                    {
                      name: "bar",
                      activities: ["be grumpy", "eat bread omlet","running","sleeping"],
                      weight: 8,
                      furcolor: "white"
                    },
                   {
                     name: "foo",
                     activities: ["sleep","pre-sleep naps","walking","eating"],
                    weight: 3
                   }
                ]
            };
console.log(cat.height);
console.log(cat.weight);
//console.log(cat.height,cat.weight);
```

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

```
var cat = {
name: "Fluffyy",
height: 50,
weight: 30,
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","walking","eating"],
                        weight: 3
                }
              ]
        };
console.log(cat.name);
```

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

```
var cat = {
 name: "Fluffyy",
 height: 50,
 weight: 30,
 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.catFriends[0].activities);
console.log(cat.catFriends[1].activities);
//console.log(cat.catFriends[0].activities, cat.catFriends[1].activities);
```

4.Print the catFriends names ?

```
var cat = {
 name: "Fluffyy",
 height: 50,
 weight: 30,
 activities: ["play" , "eat cat food"],
 catFriends: [
            {
               name: "bar",
               activities: ["be grumpy", "eat bread omlet","running","sleeping"],
               weight: 8,
               furcolor: "white"
             },
            {
               name: "foo",
               activities: ["sleep","pre-sleep naps","walking","eating"],
               weight: 3
            }
          ]
        };
```

console.log(cat.catFriends[0].name);

console.log(cat.catFriends[1].name);

// console.log(cat.catFriends[0].name, cat.catFriends[1].name);

5.Print the total weight of catFriends ?

```
var cat = {
 name: "Fluffyy",
 height: 50,
 weight: 30,
 activities: ["play" , "eat cat food"],
 catFriends: [
            {
               name: "bar",
               activities: ["be grumpy", "eat bread omlet","running","sleeping"],
               weight: 8,
               furcolor: "white"
             },
            {
               name: "foo",
               activities: ["sleep","pre-sleep naps","walking","eating"],
               weight: 3
            }
          ]
        };
var totalweight =(cat.catFriends[0].weight + cat.catFriends[1].weight);


console.log(totalweight);
//console.log(cat.catFriends[0].weight + cat.catFriends[1].weight);
//console.log(cat.weight + cat.catFriends[0].weight + cat.catFriends[1].weight);
```

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

```
var cat = {
 name: "Fluffyy",
 height: "50",
 weight: "30",
 activities: ["play" , "eat cat food"],
 catFriends: [
            {
               name: "bar",
               activities: ["be grumpy", "eat bread omlet","running","sleeping"],
               weight: 8,
               furcolor: "white"
             },
            {
               name: "foo",
               activities: ["sleep","pre-sleep naps","walking","eating"],
               weight: 3
            }
          ]
        };


console.log(cat.activities.length + cat.catFriends[0].activities.length +
cat.catFriends[1].activities.length);
```

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

```
var cat = {
 name: "Fluffyy",
 height: "50",
 weight: "30",
 activities: ["play" , "eat cat food"],
 catFriends: [
              {
                 name: "bar",
                 activities: ["be grumpy", "eat bread omlet","running","sleeping"],
                 weight: 8,
                 furcolor: "white"
               },
              {
                 name: "foo",
                 activities: ["sleep","pre-sleep naps"],
                 weight: 3
              }
           ]
         };
(cat.catFriends[0].activities).push("running", "sleeping");
(cat.catFriends[1].activities).push("playing", "eating");
console.log(cat.catFriends[0].activities);
console.log(cat.catFriends[1].activities);
// console.log(cat.catFriends[0].activities , cat.catFriends[1].activities);
```

8.Update the fur color of bar ?

```
var cat = {
 name: "Fluffyy",
 height: "50",
 weight: "30",
 activities: ["play" , "eat cat food"],
 catFriends: [
              {
                 name: "bar",
                 activities: ["be grumpy", "eat bread omlet","running","sleeping"],
                 weight: 8,
                 furcolor: "white"
               },
              {
                 name: "foo",
                 activities: ["sleep","pre-sleep naps"],
                 weight: 3
              }
           ]
         };
cat.catFriends[0].furcolor = ("cameo");

console.log(cat.catFriends[0].furcolor);
```

9. Loop over the accidents array. Change at Fault for Accident from true to false. ?

```
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"
                        }
                  ]
              };
myCar.accidents[0].atFaultForAccident = ("false");
myCar.accidents[1].atFaultForAccident = ("false");
myCar.accidents[2].atFaultForAccident = ("false");

console.log(myCar.accidents[0].atFaultForAccident,myCar.accidents[1].atFaultFor
Accident,myCar.accidents[2].atFaultForAccident);

// console.log(myCar.accidents[0].atFaultForAccident);
// console.log(myCar.accidents[1].atFaultForAccident);
// console.log(myCar.accidents[2].atFaultForAccident);
```

10.Print the dated of my accidents

```
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"
                   }
               ]
                  };
console.log(myCar.accidents[0].date);
console.log(myCar.accidents[1].date);
console.log(myCar.accidents[2].date);
//console.log(myCar.accidents[0].date,myCar.accidents[1].date,myCar.accidents[2].date);
```

Problem 1:

Parsing a JSON object's Values:

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

Input (Object):

var object = {name: "RajiniKanth", age: 33, hasPets : false};

Output:

["RajiniKanth", 33, false]

Ans;

```
var obj = {
name: "RajiniKanth",
age: 33,
hasPets : false
          };
            function printAllValues(obj){
            var newArr=Object.values(obj);
            console.log(newArr);
                                          }
printAllValues(obj);
```

OUTPUT; ['RajiniKanth', 33, false]

//

```
var obj = {
            name: "RajiniKanth",
            age: 33,
            hasPets : false
        };
console.log(obj);
```

OUTPUT; { name: 'RajiniKanth', age: 33, hasPets: false }

Problem 2:

Parsing a JSON object's Keys:

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

Example Input:

{name : 'RajiniKanth', age : 25, hasPets : true}

Example Output:

['name', 'age', 'hasPets']

Ans;

```
var obj = {
            name: "RajiniKanth",
            age: 33,
            hasPets : false
          };
function printAllKeys(obj){
        var newArr=Object.keys(obj);
console.log(newArr); }
printAllKeys(obj);
```

OUTPUT; ['name', 'age', 'hasPets']

Problem 3;

Parsing a JSON object and convert it to a list:

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

Input (Object):

var object = {name: “ISRO”, age: 35, role: “Scientist”};

Output:

[[“name”, “ISRO”], [“age”, 35], [“role”, “Scientist”]]

Ans;

```
var obj = {
            name: "ISRO",
            age: 35,
            role : "scientist"
          };
function convertObjectToList(obj){
        var newArr=Object.entries(obj);
console.log(newArr);}
convertObjectToList(obj);
```

OUTPUT; [['name', 'ISRO'], ['age', 35], ['role', 'scientist']]

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.

Input (Array):

var array = ["GUVI", "I", "am", "Geek"];

Output:

var object = {

GUVI : "Geek"

}

Ans;

```
var arr = ["GUVI", "I", "am", "Geek"];
var obj={};
function transformFirstAndLast(arr)
    {
    obj[arr[0]]=arr[3];
    console.log(obj);
    }
    transformFirstAndLast(arr);
```

Output: { GUVI: 'Geek' }

Problem 5:

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]];

Output:

var object = {
make : “Ford”
model : “Mustang”,
year : 1964
}

Ans

```
var array = [['make', 'Ford'], ['model', 'Mustang'], ['year', 1964]];
var newObj={};
function fromListToObject(arr)
{
        for (var i=0;i<arr.length;i++){
        for (var j=0;j<arr[i].length;j=j+2){
        newObj[arr[i][j]]=arr[i][j+1]
} }
           console.log(newObj) }
           fromListToObject(array);
```

Output:

{ make: 'Ford', model: 'Mustang', year: 1964 }

Problem 6:

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.

Input (Array):
var array = [[[“firstName”, “Vasanth”], [“lastName”, “Raja”], [“age”, 24], [“role”, “JSWizard”]], [[“firstName”, “Sri”], [“lastName”, “Devi”], [“age”, 28], [“role”, “Coder”]]];
Output:
[
{firstName: “Vasanth”, lastName: “Raja”, age: 24, role: “JSWizard”},
{firstName: “Sri”, lastName: “Devi”, age: 28, role: “Coder”}
]

Ans

```
var array = [[
                ["firstName", "Vasanth"],
                ["lastName", "Raja"],
                ["age", 24],
                ["role", "JSWizard"]
],
[
```

```
                ["firstName", "Sri"],
                ["lastName", "Devi"],
                ["age", 28],
                ["role", "Coder"]
]
];
var newArr=[];
function transformGeekData (arr){
        for (var i=0;i<arr.length;i++){
                let newObj={};
        for (var j=0;j<arr[i].length;j++){
        for (var k=0;k<arr[i][j].length;k=k+2){
       newObj[arr[i][j][k]]=arr[i][j][k+1];
} }
   newArr.push(newObj) }
   console.log(newArr) }
   transformGeekData (array);
```

Output:

```
[ { firstName: 'Vasanth',
    lastName: 'Raja',
    age: 24,
    role: 'JSWizard' },
  { firstName: 'Sri', lastName: 'Devi', age: 28, role: 'Coder' } ]
```

Problem 7:

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).
It is OK to use JSON.stringify().
Note: The examples below represent different use cases for the same test. In practice, you should never have multiple tests with the same name.

1)Success Case:

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

assertObjectsEqual(actual, expected, ‘detects that two objects are equal’);

Output:

Passed

2)Failure Case:

Input:

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

var actual = {foo: 5, bar: 6}

assertObjectsEqual(actual, expected, ‘detects that two objects are equal’);

Output:

FAILED

[my test] Expected {“foo”:6,”bar”:5}, but got {“foo”:5,”bar”:6}

ANS

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

function assertObjectsEqual(actual, expected, testName){
   if (JSON.stringify(expected)===JSON.stringify(actual)){
console.log("Passed") ;
       }
```

```
else { console.log("Failed") }
}
assertObjectsEqual(actual, expected,"detects that two objects are equal");
```

Output:

```
Passed
```

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

function assertObjectsEqual(actual, expected, testName){
   if (JSON.stringify(expected)===JSON.stringify(actual)){
console.log("Passed") ;
      }
else { console.log("Failed") }
}
assertObjectsEqual(actual, expected,"detects that two objects are equal");
```

Output:

```
Failed
```

Problem 8:

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) {

 // your code here return true or 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
```

Ans;

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

}

console.log(chksecurityQuestions(securityQuestions, "What was your first pet's name?","FlufferNutter"));

console.log(chksecurityQuestions(securityQuestions, "What was your first pet's name?","Fluffer"));

OUTPUT

True
False

Problem 9:

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)
{}console.log(returnMinors(students));
```

Ans

```
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){
    let list=[];
    for (var i=0;i<students.length;i++){
        if (students[i].age<20){
            list.push(students[i].name);


        }


    }
    console.log(list);
    }
returnMinors(students);
```

Output:

```
[ 'Maari', 'Bhallala Deva', 'Baahubali', 'Chulbul Pandey' ]
```