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

1. Problem 0 : Part A (15 mins):

Playing with JSON object’s Values:

Fluffy sorry, Fluffyy is my fav cat and it has 2 catFriends. Write a code to get the below details of Fluffyy so that I can take him to vet.

Solution ::

//Data Provided in Question

var cat = {

name: "Fluffy",

activities: ["play", "eat cat food"],

catFriends: [

{

name: "bar",

activities: ["be grumpy", "eat bread omblet"],

weight: 8,

furcolor: "white"

},

{

name: "foo",

activities: ["sleep", "pre-sleep naps"],

weight: 3

}

]

}

// Add height and weight to Fluffy

cat.height = 5;

cat.weight = 7;

console.log(cat.height, cat.weight)

// Fluffy name is spelled wrongly. Update it to Fluffyy

cat.name = "Fluffyy";

console.log(cat.name)

// List all the activities of Fluffyy’s catFriends.

for (var i in cat.catFriends){

console.log(cat.catFriends[i].activities);

}

// Print the catFriends names.

for (var i in cat.catFriends){

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

}

// Print the total weight of catFriends

let totalWeight = 0;

for (var i in cat.catFriends){

if (cat.catFriends[i].weight){

totalWeight += parseInt(cat.catFriends[i].weight);

}

}

console.log(totalWeight);

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

let totalActivites = 0;

totalActivites += cat.activities.length;

for (let i in cat.catFriends){

totalActivites += cat.catFriends[i].activities.length;

}

console.log(totalActivites);

// Add 2 more activities to bar & foo cats

cat.catFriends[0].activities[cat.catFriends[0].activities.length] = "abc";

cat.catFriends[0].activities[cat.catFriends[0].activities.length] = "pqr";

cat.catFriends[1].activities[cat.catFriends[0].activities.length] = "nop";

cat.catFriends[1].activities[cat.catFriends[0].activities.length] = "hlm";

// Update the fur color of bar

cat.catFriends[0].fur = "Yellow";

## ****Parsing an JSON object’s Values:****

Write a function called “printAllValues” which returns an newArray of all the input object’s values.

Input (Object):

var object = {name: “RajiniKanth”, age: 33, hasPets : false};  
Output:

[“RajiniKanth”, 33, false]

Solution ::

var printAllValues = function(inputObj){

return Object.values(inputObj);

////Another approach

//let tempArray = []

// let index=0;

// for (let i in inputObj){

// tempArray[index] = inputObj[i];

// index++;

// }

// return tempArray;

};

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

console.log(printAllValues(object));

# 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.

Example Input:  
{name : ‘RajiniKanth’, age : 25, hasPets : true}  
Example Output:  
[‘name’, ‘age’, ‘hasPets’]

Solution ::

var printAllKeys = function(inputObj){

return Object.keys(inputObj);

};

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

console.log(printAllKeys(object));

# 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.  
Input (Object):  
var object = {name: “ISRO”, age: 35, role: “Scientist”};  
Output:  
[[“name”, “ISRO”], [“age”, 35], [“role”, “Scientist”]]

Solution ::

var convertObjectToList = function(inputObj){

let toReturn = [];

let itter = 0;

for (let i in inputObj){

toReturn[itter] = [i,inputObj[i]];

itter++;

}

return toReturn

};

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

console.log(convertObjectToList(object));

# Problem 4( 5 mins):

## 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”  
}

Solution ::

var transformFirstAndLast = function(arr){

let returnObj = {};

let key = arr[0];

returnObj[key] = arr[arr.length-1];

return returnObj;

};

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

console.log(transformFirstAndLast(arr));

# 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]];  
Output:  
var object = {  
make : “Ford”  
model : “Mustang”,  
year : 1964  
}

Solution ::

var fromListToObject = function(arr){

let toReturnObj = {};

for (let i=0; i<arr.length; i++){

toReturnObj[arr[i][0]] = arr[i][1];

}

return toReturnObj;

};

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

console.log(fromListToObject(arr));

# 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.

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”}  
]

Solution ::

var transformGeekData = function(array){

//WE will be modifing the original list.

let tempDict = {};

let tempKey = "";

for(let i=0; i<array.length; i++){

let len = array[i].length;

tempDict = {};

for(let j=0; j<len; j++){

tempKey = array[i][j][0];

// console.log(array[i][j][0], array[i][j][1], tempKey);

tempDict[tempKey] = array[i][j][1];

}

array[i] = tempDict;

}

return array;

};

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

console.log(transformGeekData(array));

# 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).  
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.  
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  
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}

Solution ::

var assertObjectsEqual=function(actual, expected, testName){

actual = JSON.stringify(actual);

expected = JSON.stringify(expected);

if (actual === expected){

return "Passed";

}

else{

return `FAILED ${testName} Expected ${expected}, but got ${actual}`;

}

};

var expected = {foo: 5, bar: 1};

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

var testName = 'detects that two objects are equal'

console.log(assertObjectsEqual(actual, expected, testName));

# 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

Solution ::

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=0; i<securityQuestions.length; i++){

if (JSON.stringify(securityQuestions[i].question) === JSON.stringify(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';

status = chksecurityQuestions(securityQuestions, ques, ans);

console.log(status); // flase

# Problem 9(20 mins):

## Parsing JSON objects and Compare:

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

Solution ::

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}

];

// console.log(students)

function returnMinors(arr){

let toReturnArr=[];

let counter = 0;

for(let i=0; i < arr.length; i++){

// console.log(arr[i])

if (parseInt(arr[i].age) < 20){

toReturnArr[counter] = arr[i].name;

counter++;

}

}

return toReturnArr

}

console.log(returnMinors(students));