**[JSON Task](https://medium.com/@reach2arunprakash/guvi-zen-code-sprint-javascript-practice-problems-in-json-objects-and-list-49ac3356a8a5)**

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.

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  
 }  
 ]  
}console.log(cat);

**Basic Tasks to play with JSON**

1. Add height and weight to Fluffy
2. Fluffy name is spelled wrongly. Update it to Fluffyy
3. List all the activities of Fluffyy’s catFriends.
4. Print the catFriends names.
5. Print the total weight of catFriends
6. Print the total activities of all cats (op:6)
7. Add 2 more activities to bar & foo cats
8. Update the fur color of bar

**Code :**

var cat = {

name: 'Fluffy',

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

catFriends: [

{

name: 'bar',

activities: ['be grumpy', 'eat bread omlet'],

weight: 8,

furcolor: 'white'

},

{

name: 'foo',

activities: ['sleep', 'pre-sleep nap'],

weight: 3

}

]

}

let totalWeightOfCatFriends = 0;

//Task -1 Add height and weight to Fluffy

cat.height = '23 cm';

cat.weight= '3.5 Kg';

//TAsk -2 Fluffy name is spelled wrongly. Update it to Fluffyy

cat.name = 'Fluffyy';

//Task -3 List all the activities of Fluffyy’s catFriends.

for(const catFriendIndex in cat.catFriends)

{

console.log('\nActivities of ',cat.catFriends[catFriendIndex].name)

let activities = cat.catFriends[catFriendIndex].activities;

for(const activity in activities)

{

console.log(activities[activity]);

}

}

//Task -4 Print the catFriends names

for(const catFriendIndex in cat.catFriends){

console.log(cat.catFriends[catFriendIndex].name)

}

//Task - 5 Print the total weight of catFriends

for(const catFriendIndex in cat.catFriends){

totalWeightOfCatFriends+=cat.catFriends[catFriendIndex].weight;

}

console.log(totalWeightOfCatFriends);

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

for(const catActivity in cat.activities)

{

console.log(cat.activities[catActivity]);

}

for(const catFriendIndex in cat.catFriends)

{

let activities = cat.catFriends[catFriendIndex].activities;

for(const activity in activities){

console.log(activities[activity]);

}

}

//Task - 7 Add 2 more activities to bar & foo cats

cat.catFriends[0].activities.splice(2,0,'Drink Water','Be lazy');

cat.catFriends[1].activities.splice(2,0,'eat snacks','drink milk');

//Task - 8 Update the fur color of bar

cat.catFriends[0].furcolor = 'Black';

**Output :**

##### Output:

Task 3 - List all the activities of Fluffyy’s catFriends.

Activities of bar

be grumpy

eat bread omlet

Activities of foo

sleep

pre-sleep nap

Task 4 - Print the catFriends names

bar

foo

Task 5 -Print the total weight of catFriends

11

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

play

eat cat food

be grumpy

eat bread omlet

sleep

pre-sleep nap

##### Execution Time:

0.073s

##### Memory Used:

8356kb

# Problem 0 : Part B (15 mins):

## Iterating with JSON object’s Values

Above is some information about my car. As you can see, I am not the best driver.  
I have caused a few accidents.  
Please update this driving record so that I can feel better about my driving skills.

1. Loop over the accidents array. Change atFaultForAccident 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

}

]

}

//Change atFaultForAccident from true to false.

for(const index in myCar.accidents)

{

myCar.accidents[index].atFaultForAccident = false;

}

//2. Print the dated of my accidents

for(const index in myCar.accidents)

{

console.log(myCar.accidents[index].date);

}

console.log(myCar);

# ****Problem 1 (5 mins):****

## ****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]

**Sample Function proto:**

var obj = {name : “RajiniKanth”, age : 33, hasPets : false};function printAllValues(obj) {  
 // your code here  
}

**Code :**

var obj = {

name : 'RajiniKanth',

age : 33,

hasPets : false

};

function printAllValues(obj) {

return Object.values(obj);

}

console.log(printAllValues(obj));

**Output :**

##### Output:

[ 'RajiniKanth', 33, false ]

##### Execution Time:

0.073s

# 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’]

**Sample Function proto:**

function printAllKeys(obj) {  
 // your code here  
}

**Code :**

var obj = {

name : 'RajiniKanth',

age : 33,

hasPets : false

};

function printAllKeys (obj) {

return Object.keys(obj);

}

console.log(printAllValues(obj));

**Output :**

##### Output:

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

##### Execution Time:

0.073s

# 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”]]

**Sample Function proto:**

var obj = {name: “ISRO”, age: 35, role: “Scientist”};  
function convertListToObject(obj) {  
 // your code here  
}

# Code :

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

function convertListToObject(obj) {

return Object.entries(object);

}

console.log(convertListToObject(object));

# Output :

[ [ 'name', 'ISRO' ], [ 'age', 35 ], [ 'role', 'Scientist' ] ]

##### Execution Time:

0.075s

# 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”  
}

**Sample Function proto:**

var arr = [“GUVI”, “I”, “am”, “a geek”];function transformFirstAndLast(arr) {  
   
 return newObject;  
}

# Code :

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

function transformFirstAndLast(arr) {

let newObject ={}

newObject[arr[0]]=arr[arr.length-1];

return newObject;

}

console.log(transformFirstAndLast(arr));

# Output :

{ GUVI: 'a geek' }

##### Execution Time:

0.097s

# 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  
}

**Sample Function proto:**

var arr = [[“make”, “Ford”], [“model”, “Mustang”], [“year”, 1964]];function fromListToObject(arr) {  
 var newObject = {};  
   
 return newObject;  
}

# Code :

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

function fromListToObject(arr) {

var newObject = {};

for(const i in arr)

{

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

}

return newObject;

}

console.log(fromListToObject(arr));

# Output :

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

##### Execution Time:

0.073s

# 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”}  
]

**Sample Function proto:**

var arr= [[[“firstName”, “Vasanth”], [“lastName”, “Raja”], [“age”, 24], [“role”, “JSWizard”]], [[“firstName”, “Sri”], [“lastName”, “Devi”], [“age”, 28], [“role”, “Coder”]]];function transformEmployeeData(arr) {  
 var tranformEmployeeList = [];  
   
 //Your code  
   
 return tranformEmployeeList;  
}

# Code :

var arr = [

[

['firstName', 'Vasanth'],

['lastName', 'Raja'],

['age', 24],

['role', 'JSWizard'],

],

[

['firstName', 'Sri'],

['lastName', 'Devi'],

['age', 28],

['role', 'Coder'],

],

];

function transformEmployeeData(arr)

{

var tranformEmployeeList = [];

for (const i in arr)

{

var newObject = {};

let subArr = arr[i];

for(const j in subArr)

{

newObject[subArr[j][0]] = subArr[j][1];

}

tranformEmployeeList.push(newObject);

}

return tranformEmployeeList;

}

console.log(transformEmployeeData(arr));

# Output :

[

{ firstName: 'Vasanth’, lastName: 'Raja', age: 24, role: 'JSWizard' },

{ firstName: 'Sri', lastName: 'Devi', age: 28, role: 'Coder' }

]

##### Execution Time:

0.074s

# 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}

var expected = {foo: 5, bar: 6};  
var actual = {foo: 5, bar: 6}function assertObjectsEqual(actual, expected, testName){  
 // your code here  
}

# Code :

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

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

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

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

function assertObjectsEqual(actual, expected, testName) {

var actualValues = JSON.stringify(actual);

var expectedValues = JSON.stringify(expected);

if (actualValues === expectedValues) {

console.log('Passed');

}

else {

console.log( 'FAILED [' + testName + '] Expected ' + expectedValues + ', but got ' + actualValues );

}

}

assertObjectsEqual(actual, expected, 'My Test');

# 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) {  
  
 // 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

# Code :

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',

},

];

//Parsing JSON onbjects and compare

function chksecurityQuestions(securityQuestions, ques, ans) {

for (const i in securityQuestions) {

if (

securityQuestions[i].question !== ques ||

securityQuestions[i].expectedAnswer !== ans

) {

return false;

}

return true;

}

}

//True testcase

var ques = 'What was your first pet’s name?';

var ans = 'FlufferNutter';

//False testcase

// var ques = 'What was your first pet’s name?';

// var ans = 'DufferNutter';

var status = chksecurityQuestions(securityQuestions, ques, ans);

console.log(status);

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

# Code :

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 minor = [];

for (const i in arr) {

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

minor.push(arr[i].name);

}

}

return minor;

}

console.log(returnMinors(students));