Task – 10-07-2021

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)

1.Add height and weight to Fluffy

Answer:

cat.height=3;

cat.weight=5;

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

Answer:

  cat.name=”Fluffyy”

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

Answer:

cat.activities.forEach(activity => {

    console.log(activity)

  });

4. Print the catFriends names.

Answer:

cat.catFriends.forEach(friend => {

    console.log(friend.name)

  });

5. Print the total weight of catFriends

Answer:

var catfriendweight=0;

cat.catFriends.forEach(friend => {

  catfriendweight=catfriendweight+friend.weight

 });

 console.log(catfriendweight);

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

Answer:

var all\_activites=[];

 cat.activities.forEach(activity => {

  all\_activites.push(activity)

 });

 cat.catFriends.forEach(friend => {

   friend.activities.forEach(activity => {

     all\_activites.push(activity)

   });

 });

7.Add 2 more activities to bar & foo cats

Answer:

cat.catFriends.forEach(friend => {

    friend.activities.push("Grooming");

    friend.activities.push("Exploring")

  });

8.Update the fur color of bar

Answer:

cat.catFriends[0].furcolor="brown";

**Problem 0 : Part B (15 mins):**

Iterating with JSON object’s Values

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  
}  
]  
}

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

Answer:

myCar.accidents.forEach(accident => {

    accident.atFaultForAccident=false;

    console.log(accident);

  });

2.Print the dated of my accidents

Answer:

myCar.accidents.forEach(accident => {

    console.log(accident.date);

  });

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

Answer:

function printallvalues(params)

{

  var arr= Object.values(params);

  console.log(arr);

}

4.Write a function called “printAllKeys” which returns an newArray of all the input object’s keys.

Answer:

function printallKeys(params)

{

  var arr= Object.keys(params);

  console.log(arr);

}

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

Answer:

function convertObjectToList(params)

{

  var arr= Object.entries(params);

  console.log(arr);

}

6.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):

Answer:

function transformFirstAndLast(array)

    {

    obj[array[0]]=array[array.length-1]

    console.log(obj);

    }

7.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.

Answer:

function  fromListToObject{

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

    var obj={};

    array.forEach(subarray => {

      obj[subarray[0]]=subarray[1]

    });

    }

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

Answer:

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

  [["firstName", "Sri"], ["lastName", "Devi"], ["age", 28], ["role", "Coder"]]];

function transformEmployeeData(arr) {

 var tranformEmployeeList = [];

arr.forEach(subarray=>{

  var ob={};

subarray.forEach(elements=>{

ob[elements[0]]=elements[1]

});

tranformEmployeeList.push(ob);

});

 return tranformEmployeeList;

}

9.Write an “assertObjectsEqual” function from scratch.

Assume that the objects in question contain only scalar values (i.e., simple values like strings or numbers).

Answer:

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 [my test] Expected ",JSON.stringify(expected), ", but got", JSON.stringify(actual));

  }

}

10.Parsing JSON objects and Compare:

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”  
}  
]

Answer:

 function chksecurityQuestions(securityQuestions,quest,ans) {

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

        var secquestion=securityQuestions[i].question;

        var obj={"question":quest,"expectedAnswer":ans};

       if(secquestion=quest)

       {

           if(JSON.stringify(securityQuestions[i])==JSON.stringify(obj))

           return true;

       }

       return false;

    }

}

11.Parsing JSON objects and Compare:

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

Answer

function returnMinors(st)

   {

       var minors=[];

       students.forEach(student => {

           if(student.age<=20)

           {

               minors.push(student.name);

           }

       });

       return minors;

   }