Batch 19WD

Assignment 1: Please click on the links below for the corresponding code

[Task1](#_TASK1)

[Task2](#_Task2)

[Task3](#_Task3)

[Task4](#_Task4)

[Task5](#_Task5)

[Task6](#_Task6)

[Task7](#_Task7)

[Task8](#_Task8)

[Task9](#_Task9)

# TASK1

/\*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};

function printAllValues(obj) {

// your code here

let res = [];

for (let key in obj){

res.push(obj[key]);

}

console.log(res);

}

printAllValues(object);

/\*

Output:

["RajiniKanth", 33, false]\*/

# Task2

/\*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’]\*/

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

function printAllValues(obj) {

// your code here

let res = [];

for (let key in obj){

res.push(key);

}

console.log(res);

}

printAllValues(object);

/\*

Output:

Example Output:

[‘name’, ‘age’, ‘hasPets’]\*/

# Task3

/\*

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

}

function printAllValues(obj) {

// your code here

let res = [];

let new\_res=[];

for (let key in obj){

res=[];

res.push(key);

res.push(obj[key]);

new\_res.push(res);

}

console.log(new\_res);

}

printAllValues(obj);

/\*

Output:

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

Sample Function proto:

\*/

# Task4

/\*

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"

}

\*/

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

function transformFirstAndLast(arr)

{

let size = arr.length-1;

let newObject={};

let name = (arr[0]);

let value = arr[size];

newObject[name]= value;

//console.log(name);

return newObject;

}

console.log(transformFirstAndLast(arr));

/\*

Output:

var object = {

GUVI : "Geek"

}

\*/

# Task5

/\*

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

}

\*/

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

function fromListToObject(arr) {

var newObject = {};

for (let i in arr){

console.log(arr[i][0],arr[i][1]);

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

}

return newObject;

}

console.log(fromListToObject(arr));

/\*

Output:

var object = {

make : "Ford"

model : "Mustang",

year : 1964

}

\*/

# Task6

/\*

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"}

]

\*/

var arr= [

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

];

function transformEmployeeData(arr) {

var tranformEmployeeList = [];

var res=[];

var obj={};

var res1 = [];

for (let i in arr)

{

let newObject={};

for (let j= 0;j<arr[i].length;j++)

{

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

}

tranformEmployeeList[i] = newObject;

//res.push(newObject);

}

//console.log(res);

//return res;

return tranformEmployeeList;

}

let ans = transformEmployeeData(arr);

console.log(ans);

/\*

[

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

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

]

\*/}

let ans = transformEmployeeData(arr);

console.log(ans);

/\*

[

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

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

]

\*/

# Task7

/\*

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:

\*/

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

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

function assertObjectsEqual(actual, expected, testName){

// your code here

let exp={};

let act={};

exp= JSON.stringify(expected);

act= JSON.stringify(actual);

if (exp===act)

console.log("Passed");

else

console.log("Failed");

}

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

/\*

Output:

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

\*/

# Task8

/\*

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,answer) {

// your code here

//console.log(securityQuestions,question,answer);

for (let q in securityQuestions)

{

if (question === securityQuestions[q].question && answer === securityQuestions[q].expectedAnswer)

{//console.log("yes");

return true;

}

else

{

//console.log("no");

return false;

}

}

}

//Test case1:

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

var ans = "FlufferNutter";

var status0 = chksecurityQuestions(securityQuestions, ques, ans);

console.log(status0); // true

//Test case2:

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

ans = "DufferNutter";

status = chksecurityQuestions(securityQuestions, ques, ans);

console.log(status); // flase

# Task9

/\*

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)

{

let res=[];

for (let j in arr){

if (arr[j].age<20)

res.push(arr[j].name);

}

//console.log(res);

return res;

}

console.log(returnMinors(students));