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

**Code –**

let obj ={name:"Rajinikanth",age:33,hasPets:false};

function printAllValues()

{

console.log(Object.values(obj));

}

printAllValues();

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

**Code –**

let obj ={name:"Rajinikanth",age:33,hasPets:false};

function printAllValues()

{

console.log(Object.keys(obj));

}

printAllValues();

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

**Code** -

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

function convertListToObject(obj) {

return Object.entries(obj);

}

console.log(convertListToObject(object));

5 ) 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.

**Code -**

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

function fromListToObject(arr) {

var newObject = {};

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

let key = arr[i][0];

let value = arr[i][1];

newObject[key] = value;

}

return newObject;

}

console.log(fromListToObject(array));

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

**Code –**

var array = [

[

['firstName', 'Vasanth'],

['lastName', 'Raja'],

['age', 24],

['role', 'JSWizard']

],

[

['firstName', 'Sri'],

['lastName', 'Devi'],

['age', 28],

['role', 'Coder']

]

];

function transformEmployeeData(arr) {

var tranformEmployeeList = [];

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

let newObject = {};

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

let key = arr[i][0];

let value = arr[i][1];

newObject[key] = value;

}

tranformEmployeeList.push(newObject);

}

return tranformEmployeeList;

}

console.log(transformEmployeeData(array));

7 ) Write an “assertObjectsEqual” function from scratch.  
Assume that the objects in question contain only scalar values (i.e., simple values like strings or numbers).

**Code –**

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

var expected1 = {foo: 6, bar: 5};

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

function assertObjectsEqual(actual, expected, testName){

actualStr = JSON.stringify(actual)

expectedStr = JSON.stringify(expected)

if(actualStr == expectedStr){

return "Passed"

} else{

return "FAILED ["+testName+"] Expected "+actualStr+", but got "+expectedStr

}

}

console.log(assertObjectsEqual(actual, expected, 'test1')) console.log(assertObjectsEqual(actual, expected1, 'test2'))

8 ) 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.

**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' }

];

function chksecurityQuestions(securityQuestions,question, answer) {

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

for (keys in securityQuestions[i]){

if(keys == "question"){

if(securityQuestions[i].question == question && securityQuestions[i].expectedAnswer == answer){ return true;

}

}

}

}

return false;

}

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

var ans = 'FlufferNutter';

var status = chksecurityQuestions(securityQuestions, ques, ans); console.log(status);

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

var ans = 'DufferNutter';

var status = chksecurityQuestions(securityQuestions, ques, ans); console.log(status);

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

**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) {

var newObj = [];

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

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

newObj.push(arr[i]);

}

}

return newObj;

}

console.log(returnMinors(students));