MODULE: 2 (Data Types and Objects)

1) Write the code, one line for each action:

answer =>

1. Create an empty object user. ==>

<script>

Const user = {};

</script>

1. Add the property name with the value John. ==>

<script>

const user = {name: "John"};

document. write(user.name);

</script>

1. Add the property surname with the value Smith. ==>

<script>

Const user = {name: "John", surname: "Smith"};

document. Write (user. surname);

</script>

1. Change the value of the name to Pete. ==>

<script>

Const user = {name: "John"};

user.name = "Pete";

document. write(user.name);

</script>

1. Remove the property name from the object. ==>

<script>

const user = {name: "John", surname: "Smith"};

delete user.name;

document. write(user.name);

</script>

2)=> Is array copied?

let fruits = ["Apples", "Pear", "Orange"];

// push a new value into the "copy"

let shopping Cart = fruits; shopping Cart. push ("Banana");

// what's in fruits?

Alert (fruits. length); //?

Answer

=> Yes, It will show alert message

=> fruits => Apple, Pear, Orange, Banana

=> fruits length => 4

3)=> Map to names.

let john = {name: "John", age: 25};

let pate = {name: "Pete", age: 30};

let mary = {name: "Mary", age: 28};

let users = [ john, pate, meary];

let names = /\* ... your code \*/

alert (names); // John, Pete, Mary

answer =>

<script>

let names = users. map((item) => item.name);

</script>

4)=> Map to Object.

let john = {name: "John", surname: "Smith", id: 1};

let pate = {name: "Pete", surname: "Hunt", id: 2};

let mary = {name: "Mary", surname: "Key", id: 3};

let users = [ john, pate, mary];

let users Mapped = /\* ... your code ... \*/

/\*

Users Mapped = [

{full name: "John Smith", id: 1},

{full name: "Pete Hunt", id: 2},

{full name: "Mary Key", id: 3}

]

\*/

Alert (users Mapped [0].id) // 1

Alert (users Mapped [0]. Full name) // John Smith

answer =>

<script>

let users Mapped = users1.map((user) => ({

full name: ${user.name} $ {user. Surname},

id: user.id,

}));

</script>

5)=> Sum the properties There is a salaries object with arbitrary number of salaries. Write

the function sum Salaries (salaries) that returns the sum of all salaries using

Object. values and the for. of loop. If salaries is empty, then the result must be 0.

let salaries = {

"John": 100,

"Pete": 300,

"Mary": 250

};

Alert (sum Salaries(salaries)); // 650

answer

=> if one of salaries value is Zero Answer => 0

=> if one of salaries value is not Zero Answer => 650

<script>

let salaries = {

John: 100,

Pete: 300,

Mary: 250,

};

function sum Salaries(salaries) {

let sum = salaries. John+ salaries. Pete + salaries. Mary;

for (let Sallary of Object. entries(salaries)) {

if (salaries. John== 0 || salaries. Pete == 0 || salaries. Mary == 0) {

return document. write ("0");

} else {

return document. Write (sum);

// return alert(sum);

}

}

}

Sum Salaries(salaries);

</script>

6)=> Destructuring assignment We have an object: Write the Destructuring assignment that reads:

(a) Name property into the variable name.

(b) Year’s property into the variable age.

(c) is Admin property into the variable is Admin (false, if no such property)

(d) let user = {name: "John", years: 30};

answer =>

<script>

let user = {Name: "John Smith", Years: 35};

let {Name: name, Years: age, is Admin = false} = user;

document. Write (name);

document. Write (age);

document. Write (is Admin);

</script>

7)=> Turn the object into JSON and back Turn the user into JSON and then read it back into another variable.

user = {name: "John Smith", age: 35};

answer =>

<script>

let user\_1 = {name: "John Smith", age: 35};

let user\_2 = JSON. Parse (JSON. String (user\_1));

</script>