

MODULE: 2 (Data Types and Objects)

Question 1:- Write the code, one line for each action:

- a) Create an empty object user.
- b) Add the property name with the value John.
- c) Add the property surname with the value Smith.
- d) Change the value of the name to Pete.
- e) Remove the property name from the object.

Ans:

```
<script>
var user= {
firstName: "John",
surname: "Smith",
};
user.firstName.value = "pete";
delete user.firstName;
</script>
```

Question 2:- Is array copied?

```
let fruits = ["Apples", "Pear", "Orange"]; // push a new value into the
"copy"

let shoppingCart = fruits; shoppingCart.push("Banana"); // what's in
fruits?

alert( fruits.length ); // ?
```

Ans: The code is below

```
<script>
let fruits = ["Apples", "Pear", "Orange"];
fruitsCopy = [];
for (i = 0; i < fruits.length; i++) {
fruitsCopy[i] = fruits[i];
}
fruitsCopy.push('Mango');
let shoppingCart = fruits;
shoppingCart.push('Banana');
console.log( fruits , shoppingCart , fruitsCopy);
alert(fruits.lenght);
</script>
```

Fruits length will be 4 and shoppingCart and fruits both has a same value.

Question 9:- Map to names

```
let john = { name: "John", age: 25 };
let pete = { name: "Pete", age: 30 };
let mary = { name: "Mary", age: 28 };
let users = [ john, pete, mary ];
let names = /* ... your code */
alert( names ); // John, Pete, Mary
```

Ans:

```
<script>
let john = { name: "John", age: 25 };
let pete = { name: "Pete", age: 30 };
let mary = { name: "Mary", age: 28 };
let users = [john, pete, mary];
let names = users.map(name);
function name(item){
return [item.name].join("");

}
console.log(names);
alert(names);
</script>
```

Question 10:- Map to objects

```
let john = { name: "John", surname: "Smith", id: 1 };
let pete = { name: "Pete", surname: "Hunt", id: 2 };
let mary = { name: "Mary", surname: "Key", id: 3 };
let users = [ john, pete, mary ];
let usersMapped = /* ... your code ... */
/*
usersMapped = [
{ fullName: "John Smith", id: 1 },
{ fullName: "Pete Hunt", id: 2 },
```

```
{ fullName: "Mary Key", id: 3 }  
]  
*/  
alert( usersMapped[0].id ) // 1  
alert( usersMapped[0].fullName ) // John Smith
```

Ans:

```
let john = { name: "John", surname: "Smith", id: 1 };  
let pete = { name: "Pete", surname: "Hunt", id: 2 };  
let mary = { name: "Mary", surname: "Key", id: 3 };  
let users = [ john, pete, mary ];  
let usersMapped = users.map(user => ({  
  fullName: `${user.name} ${user.surname}`,  
  id: user.id  
}));  
alert( usersMapped[0].id ); // 1  
alert( usersMapped[0].fullName ); // John Smith
```

11. Sum the properties There is a salaries object with arbitrary number of salaries. Write the function sumSalaries(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( sumSalaries(salaries) ); // 650
```

Ans:

```
<script>
```

```
let salaries = {
```

```
  John: 100,
```

```
  Pete: 300,
```

```
  Mary: 250,
```

```
};
```

```
const ss = Object.values(salaries);
```

```
function sumSalaries(ss) {
```

```
  let sum = 0;
```

```
  for (const item of ss) {
```

```
    sum += item
```

```
  }
```

```
  alert(sum);
```

```
}
```

```
sumSalaries(ss)
```

```
</script>
```

12. 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) isAdmin property into the variable isAdmin (false, if no such property)

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

Ans:

```
let user = {
```

```
  name: "John",
```

```
  years: 30
```

```
};
```

```
let {name, years: age, isAdmin = false} = user;
```

```
alert( name ); // John
```

```
alert( age ); // 30
```

```
alert( isAdmin ); // false
```

Question 13:- 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};
```

Ans: