1. Write the code, one line for each action:
2. Create an empty object user.

Ans.let user= “ ”;

1. Add the property name with the value John

Ans**.**User.name=”Jhon”

1. Add the property surname with the value Smith.

Ans**.** user.surname = "Smith";

1. Change the value of the name to Pete.

Ans**.** user.name = "Pete";

1. Remove the property name from the object.

Ans**.** delete user.name;

1. 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 result is 4:

Input:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

  </head>

  <body>

    <script>

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

      let shoppingCart = fru;

      // Add Banana into the array

      shoppingCart.push("Banana");

      //Print Length of an Array in Alert Box

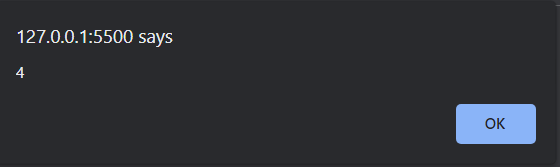
      alert(fruits.length);

    </script>

  </body>

</html>

**Output:**



1. 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.

Input:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        //Creates an array

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

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

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

        //Merge three Array

        let users = [ john, pete, mary ];

        // Maps the current Array

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

        // John, Pete, Mary in Alert Box

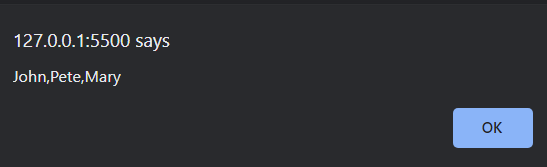
        alert( names );

    </script>

</body>

</html>

**Output:**



1. 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 ... \*/

Ans.

Input:

Example 1:

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

  </head>

  <body>

    <script>

      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);

      alert(usersMapped[0].fullName);

    </script>

  </body>

</html>

**Example 2:**

**Input:**

 <html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

  </head>

  <body>

    <script>

        //Creates an Array

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

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

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

      //Merge an Array

      let users = [john, pete, mary];

      let usersMapped = users.map((user) => ({

        Name: `${user.name} ${user.surname}`,

        id: user.id,

      }));

      //Displays an Id and Name in Alert

      alert(usersMapped[0].id);

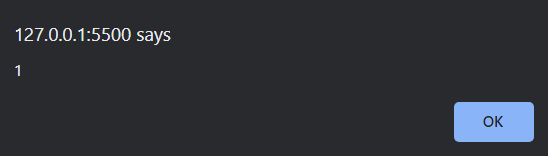
      alert(usersMapped[0].Name);

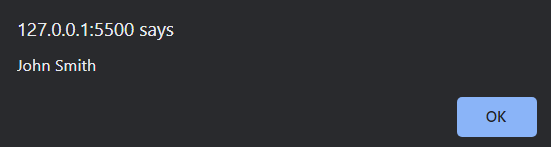
    </script>

  </body>

</html>

Output:





1. 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.

Input:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        function sumofsalaries(salaries) {

            let value = 0;

            // Put the value in for of loop for summation

                for (let salary of Object.values(salaries)) {

                value += salary;

                }

                return value;

                }

            // Creating an Array

                let salaries = {

                "John": 100,

                "Pete": 300,

                "Mary": 250

                };

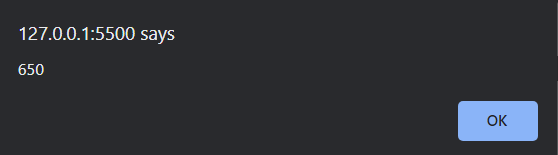
                alert( sumofsalaries(salaries) );

    </script>

</body>

</html>

**Output:**



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) isAdmin property into the variable isAdmin (false, if no such property) d) let user = { name: "John", years: 30};

Ans.

Input:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

  </head>

  <body>

    <script>

        //Cretes an Array

      let user = {

        name: "John",

        years: 30,

      };

      //Merge an Array

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

      alert(name);

      alert(age);

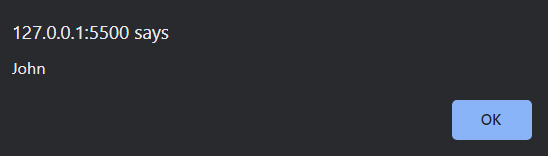
      alert(isAdmin);

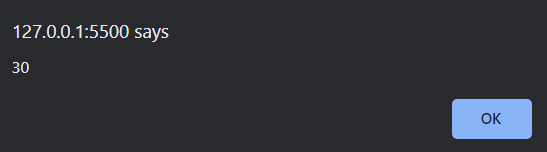
    </script>

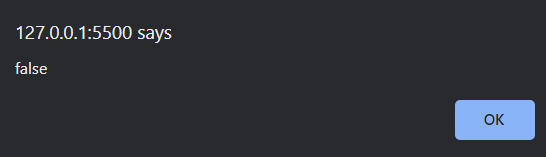
  </body>

</html>

**Output:**







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};

Ans.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <Script>

            let user = {

          name: "John Smith",

          age: 35

          };

          //Json Variable

          let json\_var = JSON.parse(JSON.stringify(user));

    </Script>

</body>

</html>