

# Module 10 :- JavaScript Object

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

A). Create an empty object user :-

```
<!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>
      var objectA={};
      var objectB=new Object();
    </script>
  </body>
</html>
```

B). Add the property name with the value John :-

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
const person = {
  firstname: "John"
};

document.getElementById("demo").innerHTML = person.firstname;
</script>

</body>
</html>
```

C). Add the property surname with the value Smith :-

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
const person =
{
  surname: "Smith"
};

document.getElementById("demo").innerHTML = person.surname;
</script>

</body>
</html>
```

D). Change the value of the name to Pete. :-

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
const person = {
  firstname: "Pete"
};

document.getElementById("demo").innerHTML = person.firstname;
</script>

</body>
</html>
```

E). Remove the property name from the object. :-

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>
```

```

<script>
var person = {
  firstname: "John",
  surname: "Smith"
};

delete person.surname;
document.getElementById("demo").innerHTML =
person.firstname + " and surname is " + person.surname;
</script>

</body>
</html>

```

## 2). Is array copied?

```

<!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>
      "use strict";

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

      let shoppingCart = fruits;

      shoppingCart.push("Banana");
      document.write(shoppingCart);
      alert(fruits.length); // 4
    </script>
  </body>
</html>

```

### 3). Map to names?

```
<!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 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(item => item.name);
    document.write(names);
    alert( names ); // John, Pete, Mary
  </script>
</body>
</html>
```

### 4). Map to objects

```
<!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 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
    }));

    /*
    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
</script>
</body>
</html>

```

**5). 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.

```

<!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 sumSalaries(salaries) {
        let sum = 0;
        for (let salary of Object.values(salaries)) {
          sum += salary;
        }

        return sum; // 650
      }

      let salaries = {
        John: 100,
        Pete: 300,

```

```

        Mary: 250,
    };

    alert(sumSalaries(salaries)); // 650
</script>
</body>
</html>

```

## 6). Destructuring assignment We have an object: Write the Destructuring assignment that reads:

A) Name property into the variable name.

```

<!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",
        years: 30,
      };

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

      alert(name); // John
      alert(age); // 30
      alert(isAdmin); // false
    </script>
  </body>
</html>

```

B) Year's property into the variable age.

```

<!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>

```

```

</head>
<body>
  <script>

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

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

    alert(name); // John
    alert(age); // 30
    alert(isAdmin); // false
  </script>
</body>
</html>

```

C). isAdmin property into the variable isAdmin (false, if no such property) :-

```

<!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",
        years: 30,
      };

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

      alert(name); // John
      alert(age); // 30
      alert(isAdmin); // false
    </script>
  </body>
</html>

```

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

```
<!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",
        years: 30,
      };

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

      alert(name); // John
      alert(age); // 30
      alert(isAdmin); // false
    </script>
  </body>
</html>
```

## 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};

```
<!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,
      };

      let user2 = JSON.parse(JSON.stringify(user));
```



```
    </script>  
  </body>  
</html>
```