-> SPREAD OPERATOR

-> It is denoted by ...(Three dots) and a very powerful feature introduced in ES6.

-> It allows you to expand an iterable objects such as ->
-> Array
-> String
-> Objects

-> into individual objects

-> It is used commonly in -> Function Calls, Array Literals and Object Literals.

-> You can use the spread operator to expand an array into individual elements.

-> This is particularly useful when you want to combine arrays or pass array elements as arguments to a function.

-> Syntax:

```
   let variablename1 = [...value];
```

Array Concatenation

```
<body>
    <p></p>
    <script>
        const arr1 = [1, 2, 5];
        const arr2 = [4, 6, 7];
        const concat = [...arr1, ...arr2]
        let p = document.querySelector("p");
        p.innerHTML = concat;
    </script>
</body>
```

Array Copy

```
<body>
```

```
    <p></p>
    <script>
        const org = [1, 2, 5];
        const cop = [...org];
        let p = document.querySelector("p");
        p.innerHTML = cop;
    </script>
</body>
------------------------------------------------------------------------

------------------------------------------------------------------------
Object Copy
------------------------------------------------------------------------
<script>
    const org = {name:"John", age:25};
    const cop = {...org};
    console.log(cop);
</script>
------------------------------------------------------------------------

------------------------------------------------------------------------
Object Merge
------------------------------------------------------------------------
 <script>
    const org1 = {a:1};
    const org2 = {b:2};
    const cop = {...org1, ...org2};
    console.log(cop);
</script>
------------------------------------------------------------------------

------------------------------------------------------------------------
Function Arguments
------------------------------------------------------------------------
<body>
    <p></p>
    <script>
        function addNumbers(a, b, c){
            return a + b + c;
        }
        const nums = [4,7,9];

        //Passing the whole array as argument to addNumbers()
        const sum = addNumbers(...nums);

        document.querySelector("p").innerHTML = sum;
    </script>
</body>
------------------------------------------------------------------------

------------------------------------------------------------------------
-> It creates a shallow copy of an object or array.
```

-> This means that while the top-level elements are copied, nested objects or arrays within the spread may still be referenced instead of deeply copied.

-> If you need to create a deepy copy, use other techniques
   -> Recursion
   -> External Libraries

-> Keep in mind that if there are duplicate properties in the objects being merged, the property from the latter object will overwrite the one from the former.