

Day 9



JavaScript Array Methods:

JavaScript Array Methods

Arrays in JavaScript come with many built-in methods to add, remove, search, and manipulate elements easily.

Example: `.toString()`

```
JS Main.js > ...
1  let arr = [1,3,4,56];
2  console.log(arr); //Array
3  let b = arr.toString();
4  console.log(b); //String
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day9> node Main.js
[ 1, 3, 4, 56 ]
1,3,4,56
```

Example: `.join()`

```
6  let arr = [1,3,4,56];
7  let b = arr.join("_");
8  console.log(b);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day9> node Main.js
1_3_4_56
```

Example: .pop()

```
JS Main.js > ...
10 let arr = [1,3,4,56];
11 console.log(arr);
12 console.log(arr.pop());
13 console.log(arr);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day9> node Main.js
[ 1, 3, 4, 56 ]
56
[ 1, 3, 4 ]
```

Example: .push()

```
JS Main.js > ...
15 let arr = [1,2,3,4];
16 console.log(arr);
17 console.log(arr.push(22));
18 console.log(arr);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day9> node Main.js
[ 1, 2, 3, 4 ]
5
[ 1, 2, 3, 4, 22 ]
```

Example: .shift()

```
JS Main.js > ...
20 let arr = [1,2,3,4];
21 console.log(arr);
22 console.log(arr.shift());
23 console.log(arr);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day9> node Main.js
[ 1, 2, 3, 4 ]
1
[ 2, 3, 4 ]
```

Example: .unshift()

```
JS Main.js > ...
25 let arr = [1,2,3,443];
26 console.log(arr);
27 console.log(arr.unshift(33));
28 console.log(arr);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day9> node Main.js
[ 1, 2, 3, 443 ]
5
[ 33, 1, 2, 3, 443 ]
```

Example: delete operator

```
JS more-methods.js > ...
1 let arr = [10,4,5,6];
2 console.log(arr);
3 console.log(arr.length)
4 delete arr[1];
5 delete arr[3];
6 console.log(arr);
7 console.log(arr.length);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
[ 10, 4, 5, 6 ]
4
[ 10, <1 empty item>, 5, <1 empty item> ]
4
```

Example:

```
9 let a1 = [1,2,3,4];
10 let a2 = [88,9,9,0,22];
11 console.log(a1.concat(a2));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
[
  1, 2, 3, 4, 88,
  9, 9, 0, 22
]
```

Example: `.concat()` doesn't change the original array

```
13 let a1 = [1,2,3,4];
14 let a2 = [88,9,9,0,22];
15 let a3 = [3,4];
16 a3.concat(a1,a2);
17 console.log(a3);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
[ 3, 4 ]
```

Example: `.sort()`

```
19 let arr = [88,44,33];
20 console.log(arr.sort());
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
[ 33, 44, 88 ]
```

Example: `.sort()`

```
22 let arr = ["a", "b", "d", "c"];
23 console.log(arr.sort());
24 let arr2 = ["Uditya", "Aditya"];
25 console.log(arr2.sort());
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
[ 'a', 'b', 'c', 'd' ]
[ 'Aditya', 'Uditya' ]
```

Example: to sort in descending order, we will use function

```
28  const compare = (a, b) => {
29    |   return b - a;
30  };
31
32  let arr = [1, 34, 5, 6, 22];
33  arr.sort(compare);
34  console.log(arr);
35
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
[ 34, 22, 6, 5, 1 ]
```

Example: .reverse()

```
36  let arr = [2,445,6,7];
37  console.log(arr.reverse());
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
[ 7, 6, 445, 2 ]
```

Example: .splice()

```
39  let fruits = ["apple", "banana", "cherry", "date"];
40
41  let removed = fruits.splice(1, 2, "kiwi", "mango");
42
43  console.log("Removed:", removed);
44  console.log("Updated array:", fruits);
--
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
Removed: [ 'banana', 'cherry' ]
● Updated array: [ 'apple', 'kiwi', 'mango', 'date' ]
```

Example: .slice()

```
47 let fruits = ["apple", "banana", "cherry", "date", "elderberry"];
48
49 let sliced = fruits.slice(1, 4);
50
51 console.log("Sliced:", sliced);
52 console.log("Original array:", fruits);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day9> node .\more-methods.js
Sliced: [ 'banana', 'cherry', 'date' ]
Original array: [ 'apple', 'banana', 'cherry', 'date', 'elderberry' ]
```

--The End--