Discuss the following array methods and have an example for each. We will do the review same way we did yesterday, You share your screen explain to us and do a demo for the same

1. push() – used to add a new item in an array:

eg.const fruits = [“banana”, “orange”, “apple”, “mango”]

fruit.push(“kiwi”);

can push 2 or more items

1. shift() method returns the shifted elements

const fruits=[“Banana”, “Orange” , “Apple”, “Mango”];

fruits.shift() ;

1. unshift – adds one or more elements to the begging of an array and returns the new length of the array.

To add new elements to an array

Const fruits= [“Bananas”, “Orange”, “Apple”, “Mango”];

Fruits.unshift(“Lemon”, “Pineapple”);

1. pop –removes the last element and returns the element it removed:

const fruits = [“Banana”, “Orange”, “Mango”];

fruits.pop();

1. length – set or returns the number of elements in an array

const fruits == [“Bananas”, “Orange”, “Apple”, “Mango”];

fruit.length =2;

1. splice -
2. reduce - example of using the **reduce()** method to calculate the sum of all elements in an array:

const numbers = [1, 2, 3, 4, 5];

const sum = numbers.reduce((accumulator, currentValue) => { return accumulator + currentValue; }, 0); console.log(sum); // Output: 15

In this example, the initial value of the accumulator is 0. The callback function takes the accumulator and adds the current value to it. The result of the callback function is then used as the accumulator in the next call to the callback function. The **reduce()** method returns the final value of the accumulator, which in this case is the sum of all elements in the array.

1. includes – returns true if an array contains a specified value an returns false if the value is not found

const fruits == [“Bananas”, “Orange”, “Apple”, “Mango”];

fruits. Includes(“Mango”);

if you want to start search at position 3 for example

fruits.includes(“Banana”, 3);

join

indexof

includes

finc

findIndex

map

filter

Fill

reverse

String methods

charAt(index)

concat() replace()

split()

substr(start, length)

substring(start,end)

slice(start, end)

toLowerCase()

toUpperCase()

trim()

includes()

search()