ARRAY METHODS IN JAVASCRIPT

1. Concat () method

- create a new array with the combination of two or more array.

Syntax :   —@iam_frontendev

```
array.concat (Val1, val2, ... , ValN);
```

EXAMPLE

```
Var chars = ["A", "B", "C"];

Var num = [10, 20, 30];

var concat_res = chars.concat (num);

console.log (concatres);
```

Output :
        A, B, C, 10, 20, 30

2. every ()

- check whether given elements are true or false in an array as provided in the condition
- Return true/ false.

SYNTAX

condition to test the
↑   function

```
Array.every (callback, thisArg)
```
↳ optional parameter

EXAMPLE

```
var vals = [150,400, 450, 375, 200];

function check_num (value) {
    return value > 100;
}

document.writenln (vals. every (check_num)
);
```

OUTPUT :

true

3. fill ()

- fills the specified ∧Static value by modifying the original values.

Syntax :-

```
array.fill (value)
```

Example :-

```
var myarray = ["England", "Australia"
                        "India"] ;
var array_result = myarray.fill ("EDUCBA");

document.writeln ( myarray ) ;
```

OUTPUT :-

EDUCBA, EDUCBA, EDUCBA

4. filter ()

The filter () method filters the array of elements based on the given condition.

SYNTAX :-

```
array. filter (callback, thisArg)
```

Example :-

```
var vals = [150, 200, 345, 375, 125];
function check_num (value)
{
   return value > 150;
}
console.log (vals.filter (check_num));
```

Output :-

200, 345, 375

JAVASCRIPT ARRAY METHODS

— @iam_frontendev

- ## toString ()

  → Return — Returns an array as a comma separated string :

  Example :-

  ```
  const fruits = ["Banana", "Orange", "Apple"];

  console.log ( fruits.toString ());
  ```

  Output :-

  Banana, Orange, Apple, Mango

- ## join ()

  → join () method joins array elements into a string

  Example :-

  ```
  const fruits = ["Banana", "Orange", "Apple"]

  console.log ( fruits.join ("*");
  ```

  Output:- Banana * Orange * Mango

- **Pop ()**

  → Removes the last element from array

  → Return - value that was popped out

  Example -

  ```
  const fruits = ["Banana", "Orange", "Apple"];

  console.log (fruits.pop());
  ```

  Output :-

  'Mange' 'Apple'

- **Push ()**

  → Adds new element to an array (at the end):

  ```
  const fruits = ["Banana", "Orange", "Apple"];

  console.log (fruits.push("kiwi"));
  ```

  Return - push () returns the new array length

  Output :-   '4' → New array length

  -@iam-frontendev

SHIFING ELEMENTS

- **Shift ()**

  → Removes the first array element and "shifts" all other elements to a lower index.

```
const fruits = ["Banana", "Orange", "Apple"];
```

```
Console.log (fruits.shift());
```

Output :-   'Banana'

Return → The **shift()** method return the value that was shifted out.

- **unshift()**

  → Add new element to an array (at the beginning), and "unshifts" older elements.

```
const fruits = ["Banana", "Orange", "Apple"]
```

```
console.log (fruits.unshift("Lemon"));
```

- Return - New Array length

  Output :-   °4 '