SUNIL VISHWAKARMA
@linkinsunil

array.map()

in JavaScript

Map takes an array and returns a new array of the same length as the original array, but with values changed based on a callback function.

optional

syntax

```
arr.map(function(element, index, array){  }, this);
```

Usages explained-

1. Double the elements of an array
2. Reformat objects in an array
3. Modify certain elements in an array
4. Convert a String to an Array
5. Iterate through a NodeList
6. Render a List in React.js

Bonus: Implemention of map method

swipe ⟶


SUNIL VISHWAKARMA
@linkinsunil

Double the elements of an array

You can create a new array from another array using the map() method. For example, you can double the elements of an integer array and construct a new array from the initial array.

```
let initialArray = [1,2,3,4,5]
let doubles = initialArray.map(x => x * 2);

console.log(initialArray);
// returns [1,2,3,4,5]

console.log(doubles);
// returns [2,4,6,8,10]
```

swipe ⟶


SUNIL VISHWAKARMA
@linkinsunil

Reformat objects in an array

You can reformat an array of objects using the map() method. For example, you have an array of objects that contains the first and last names of the people. Also, you want to create a new array containing the people's full name in the people array.

```
let lineup = [
  {
    id: 1,
    firstName: "Magic",
    lastName: "Johnson"
  }, {
    id: 2,
    firstName: "Kobe",
    lastName: "Bryant"
  }, {
    id: 3,
    firstName: "Lebron",
    lastName: "James"
  }
];
```

```
let lineupSpeech = lineup.map( player => {
 let newObj = {};
 newObj["fullName"] = player.firstName + " " + player.lastName;
 return newObj;
 })

console.log(lineup);
/*
[
  { id: 1, firstName: 'Magic', lastName: 'Johnson' },
  { id: 2, firstName: 'Kobe', lastName: 'Bryant' },
  { id: 3, firstName: 'Lebron', lastName: 'James' }
]
*/
console.log(lineupSpeech);
/*
[
  { fullName: 'Magic Johnson' },
  { fullName: 'Kobe Bryant' },
  { fullName: 'Lebron James' }
]
*/
```

swipe ⟶

Modify certain elements in an array

Instead of doubling every element in the array, you can double the specified elements. For example, you may want to double the elements if they are odd.

```
const numArray = [1, 2, 3, 4];

const sqrts = numArray.map( (num) => {
  return num % 2 === 1 ? Math.sqrt(num): num
} );
```

swipe ⟶

Convert a String to an Array

You can convert a string to an array using map() method. To do that, you can get help from the .call() method.

```
const language = "JavaScript";
const map = Array.prototype.map;

const letters = map.call(language, eachLetter => {
    return `${eachLetter}`
})

console.log(letters)
// ['J','a','v','a','S','c','r','i','p','t']
```

swipe ⟶

Iterate through a NodeList

You can use map() method to iterate through the objects collected by querySelectorAll(). It is possible since querySelectorAll() returns a NodeList.

```
let NodeList = document.querySelectorAll("p");

let values = Array.prototype.map
  .call(NodeList, function(obj) {
    return obj.value
})
```

swipe ⟶

Render a List in React.js

You can render a list in React with map().

```
import React from 'react';
import ReactDOM from 'react-dom';

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

const listItems = numbers.map((number) =>
<li> {number} </li>
);

ReactDOM.render(
 <ul>{listItems}</ul>,
 document.getElementById('root')
);
```

swipe ⟶

Bonus: Implemention of map method

Here's the implementation of map() method

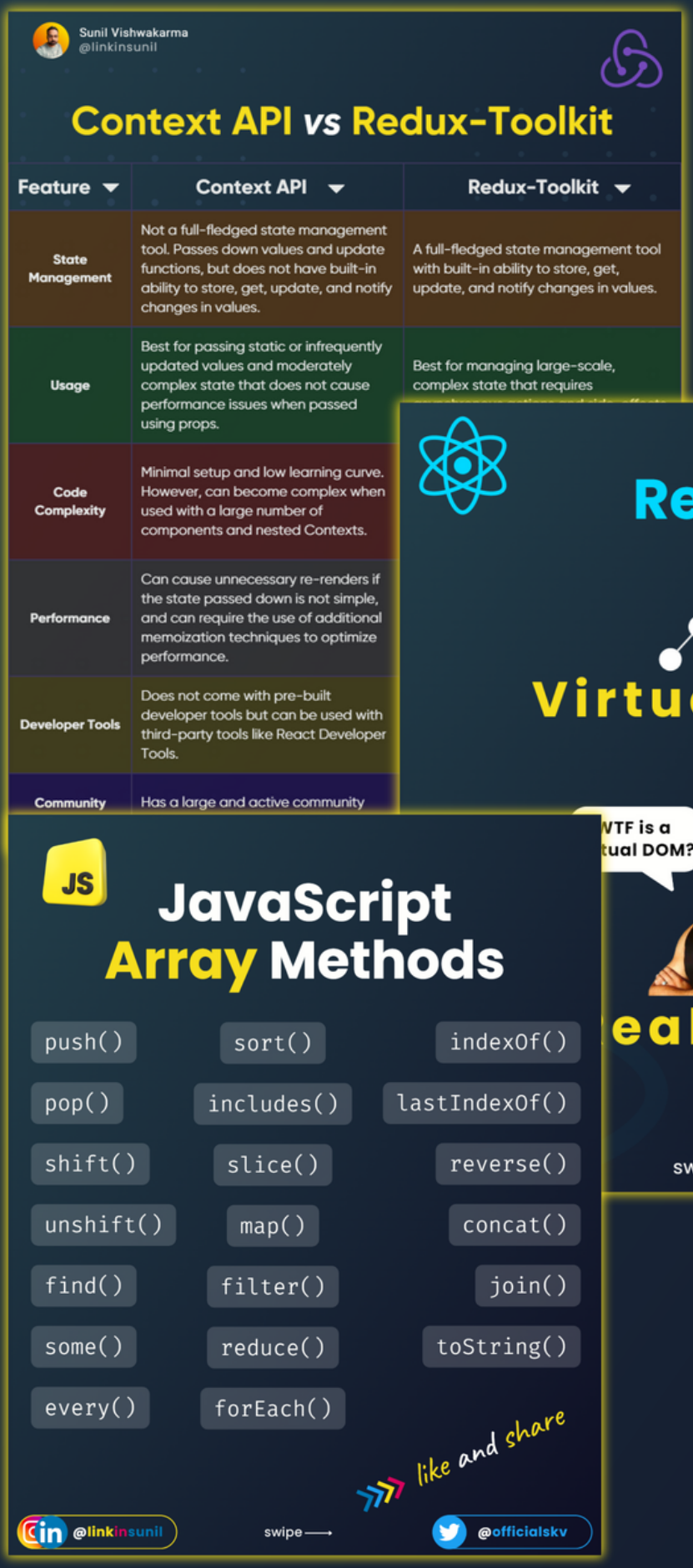
```
const arrayToBeMapped = [1, 2, 3, 4, 5, 6];

const map = (array, cb) => {
  const result = [];
  for (let i = 0; i < array.length; i++) {
      result.push(cb(array[i], i, array));
  }
  return result;
}

const mappedArray = map(arrayToBeMapped,
(number) => {
    return number * 2
})

console.log(mappedArray)
// [2, 4, 6, 8, 10, 12]
```

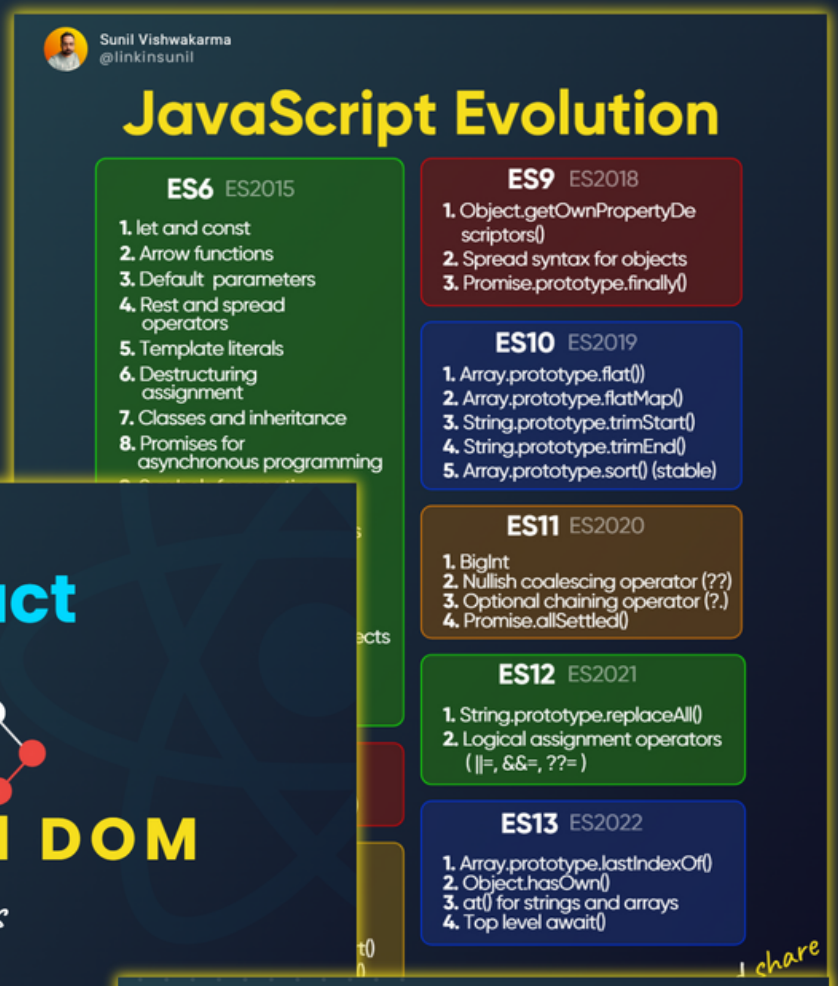
swipe ⟶

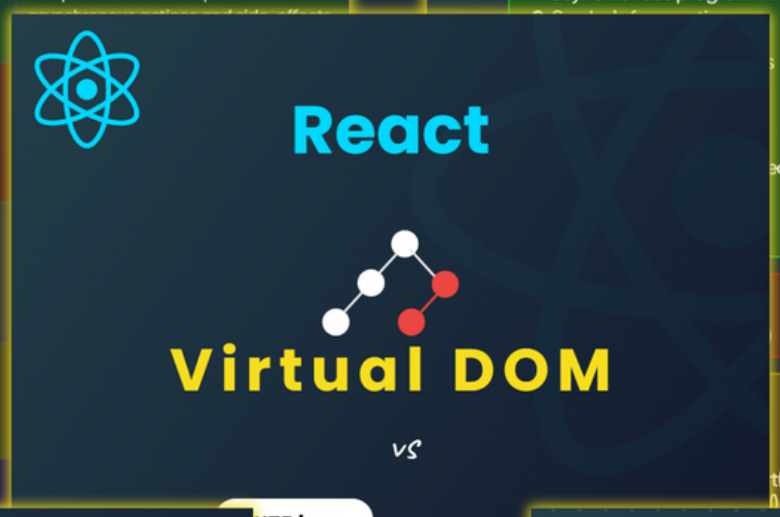
SUNIL VISHWAKARMA
@linkinsunil

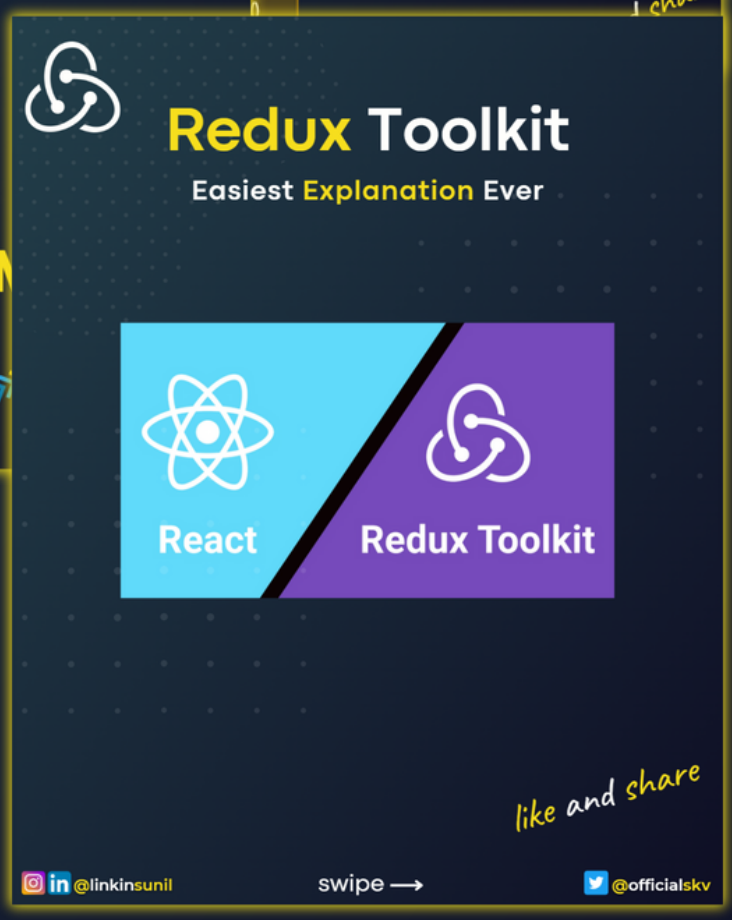
Thats a Wrap!

If you liked it, visit my profile and checkout other posts









Follow For More ⚡

Sunil Vishwakarma ✅