3 KILLER ARRAY METHODS

- MAP()
- FILTER()
- REDUCE()

Ajay Yadav
@ATechAjay

map( )

🌱 It accepts a callback function once for each array element and returns a new array.

🌱 The map( ) method doesn't execute the function for an empty array.

Syntax of map( )

```
array.map(function (item, index, arr) {
        //Lines of code
});
```

Ajay Yadav
@ATechAjay

🌱 item - It's a required parameter, that is the value of the current array item.

🌱 index - Optional, it is the current index of the array element that is being processed.

🌱 arr -Optional, the array on which the map( ) method was called.

Example of map( )

```
let arr = [10, 20, 30, 40, 50, 60]
let square = arr.map(function (item, index) {
          return `Index ${index} Square: ${item * item}`
});
console.log(square);
```

```
[ 'Index 0 Square: 100',
  'Index 1 Square: 400',
  'Index 2 Square: 900',
  'Index 3 Square: 1600',
  'Index 4 Square: 2500',
  'Index 5 Square: 3600' ]
```

🌱 The new resultant array will be square variable.

Ajay Yadav
@ATechAjay

filter( )

🌱 The filter() method returns an array that passes the condition. If no elements pass the condition, it returns an empty array.

🌱 It doesn't execute the callback function for empty elements.

Syntax of filter( )

```
array.filter(function (item, index, arr) {
        //Lines of code
});
```

Ajay Yadav
@ATechAjay

🌱 item - It's a required parameter, that is the value of the current array item.

🌱 index - Optional, it is the current index of the array element that is being processed.

🌱 arr - Optional, the array on which the filter( ) method was called.

Example of filter( )

```
const user = ["Ajay", "Sivam", "Anuj", "Shankar"];

let nameStartsWithA = user.filter(function (name) {
          return name.startsWith("A");
});

console.log(nameStartsWithA);  //[ 'Ajay', 'Anuj' ]
```

🌱 The filtered array will be nameStartsWithA variable.

Ajay Yadav
@ATechAjay

reduce( )

🌱 The reduce( ) method executes a reducer function on each element of the array.

🌱 It returns a single value as a result and doesn't execute the function for an empty array element.

Syntax of reduce( )

```
array.reduce(function (accumulator, item, index, arr) {
        //Lines of code.
}, initialValue);
```

Ajay Yadav
@ATechAjay

🌱 accumulator - It's a value of the previous function call, or the InitialValue for the first call.

🌱 If there is no any InitialValue provided then it’s accumulate the first item of the array and item is initialized to the second value in the array.

🌱 item - It's a required parameter, that is the value of the current item of the array.

🌱 index - Optional, it is the current index of the array element that is being processed.

🌱 arr - Optional, the array on which the reduce( ) method was called.

🌱 InitialValue - Optinal, It’s a starting value that is used to initialize the callback function for the first time.


Ajay Yadav
@ATechAjay

Example of reduce( )

```
const inr = [10, 20, 30, -20, 50];

const totalINR = inr.reduce(function (total, item) {
        return total + item;
}, 0);

console.log(totalINR); //90
```

🌱 For the 1st iteration:

- 🍭 total = 0(InitialValue)
- 🍭 item = 10
- 🍭 InitialValue = 0

Ajay Yadav
@ATechAjay

Example of reduce( )

```
const inr = [10, 20, 30, -20, 50];

const totalINR = inr.reduce(function (total, item) {
          return total + item;
});

console.log(totalINR); //90
```

🌱 In the above code, I've omitted the InitialValue value, so the total variable is initialised with 10 and the item variable is initialised with 20.

🌱 It returns a single value that is stored in a variable called totalINR.

NOTE: Array methods does not changed the original array.

Ajay Yadav
@ATechAjay

💙

Thanks for the reading

Follow me for ultimate content
on HTML, CSS, JavaScript,
React, DSA, and Git

Ajay Yadav

@ATechAjay