Javascript get unique items from array

Posted on March 11, 2019 | by Prashant Yadav

Posted in Arrays, Javascript

Learn how to filter unique items from the array in javascript.

We will see a different approach to filter the unique items from the simple, nested arrays as well as an array of objects.

Unique items from simple Array

Using filter()

filter() method takes a callback function in which we can provide the condition to filter the unique items. If the condition is `true` then the item will be added to the new array. It return's the new array with unique items.

Copy

```
let friends = ['prashant', 'panam', 'sachin', 'sachin', 'pranav', 'panam'];
let unique = friends.filter((e, i) => friends.indexOf(e) === i);
console.log(unique);
// ["prashant", "panam", "sachin", "pranav"]
```

indexOf() method returns the index of the first matching element in the array. So we are using it to filter the unique elements.

For Example
`"sachin"` is present at index `2` & `3` if we check this with the indexOf() method then `"sachin"` of index `3` will return `2` as it is the first matching index, so as `3 !== 2` it will not be added to unique arrays.

Practically prepare for your JavaScript interview

JavaScript Revision

JavaScript-Concept Based Problems

Data Structures

Algorithms

Machine Coding

Web Fundamentals

Using Set

Set is a new data structure added in ES6 which stores only unique items.

Copy

```
let count = [1, 2, 3, 4, 5, 1, 4, 5, 2, 6, 1, 9, 10];
let set = new Set();

for(let val of count){
   set.add(val);
}

console.log(set);
/*
Set(8) {1, 2, 3, 4, 5, …}
size: (...)
__proto__: Set
[[Entries]]: Array(8)
0: 1
1: 2
2: 3
3: 4
4: 5
5: 6
6: 9
7: 10
length: 8
*/
```

As you can see it holds only the unique items.

We can use spread ... operator to make our things more easy.

Copy

```
let count = [1, 2, 3, 4, 5, 1, 4, 5, 2, 6, 1, 9, 10];
let set = new Set(...count);
console.log(set);
/*
Set(8) {1, 2, 3, 4, 5, …}
size: (...)
__proto__: Set
[[Entries]]: Array(8)
0: 1
1: 2
2: 3
3: 4
4: 5
5: 6
6: 9
7: 10
length: 8
*/
```

Unique items from nested Array

Using JSON.stringify()

Copy

```
let arr = [[10,15], [10,14], [10,15], [7,3], [1,2], [8,9]];

let Unique = (arr) => {
    //To store the unique sub arrays
    let uniques = [];

    //To keep track of the sub arrays
    let itemsFound = {};

    for(let val of arr) {
        //convert the sub array to the string
        let stringified = JSON.stringify(val);

        //If it is already added then skip to next element
        if(itemsFound[stringified]) {
           continue;
        }

        //Else add the value to the unique list
        uniques.push(val);

        //Mark it as true so that it can tracked
        itemsFound[stringified] = true;
    }

    //Return the unique list
    return uniques;
}

let filtered = Unique(arr);
console.log(filtered);

/*
(5) [Array(2), Array(2), Array(2), Array(2), Array(2)]
0: (2) [10, 15]
1: (2) [10, 14]
2: (2) [7, 3]
3: (2) [1, 2]
4: (2) [8, 9]
length: 5
__proto__: Array(0)
*/
```

We are using JSON.stringify() to convert each inner array to string and store it in the `itemsFound`. If the `itemsFound` is false previously then we push the current inner array to unique list and mark the `itemsFound` as `true` else we skip the execution and move to the next item using `continue` statement.

Filtering array of objects

If we you want to filter based on particular property then we do so by the following method.

Copy

```
let arr = [{name:'prashant'}, {name:'sachin'}, {name:'prashant'}, {name:'pranav'}];

let Unique = (arr) => {
    //Store the unique
    let uniques = [];

    //Track the items added to the uniques
    let itemsFound = {};
    for(let val of arr) {
        //If item is already added then move to the next item
        if(itemsFound[val.name]){
          continue;
        }

        //Else push it to the unique list
        uniques.push(val);

        //Mark it as added
        itemsFound[val.name] = true;
    }

    //Return the uniques
    return uniques;
}

let filtered = Unique(arr);
console.log(filtered);

/*
[{name: "prashant"},
{name: "sachin"},
{name: "pranav"}]
*/
```

If you want filter based on the whole object then we can use the JSON.stringify() to get the unique items.

```
let arr = [{name:'prashant', age: 15}, {name:'sachin', age: 16}, {name:'prashant',
age: 16}, {name:'prashant', age: 16}, {name: 'pranav', age: 12}];

let Unique = (arr) => {
    //To store the unique sub arrays
    let uniques = [];

    //To keep track of the sub arrays
    let itemsFound = {};

    for(let val of arr) {
        //convert the sub array to the string
        let stringified = JSON.stringify(val);

        //If it is already added then skip to next element
        if(itemsFound[stringified]) {
           continue;
        }

        //Else add the value to the unique list
        uniques.push(val);

        //Mark it as true so that it can tracked
        itemsFound[stringified] = true;
    }

    //Return the unique list
    return uniques;
}

let filtered = Unique(arr);
console.log(filtered);
/*
[{name: "prashant", age: 15},
{name: "sachin", age: 16},
{name: "prashant", age: 16},
{name: "pranav", age: 12}]
*/
```

As you can see there were two `{name: "prashant", age: 16}` and we filtered it to get only unique items.

Unique properties of array of objects

We can use the spread ... along with map and Set to get the unique properties from the array of objects.

Copy

```
let arr = [{name:'prashant', age: 15}, {name:'sachin', age: 16}, {name:'prashant',
age: 16}, {name:'prashant', age: 16}, {name: 'pranav', age: 12}];

let uniqueName = [...new Set(arr.map((person) => person.name))];
console.log(uniqueName);
//["prashant", "sachin", "pranav"]

let uniqueAge = [...new Set(arr.map((person) => person.age))];
console.log(uniqueAge);
//[15, 16, 12]
```

map returns a new array with the added items.

We are passing the new array of `person.name` or `person.age` to the Set which will have the unique items and then spreading it as individual item using spread ... operator.

Recommended Posts:

- Convert Kilometers to miles in JavaScript
- Implement custom instanceof method
- Program to find an element in array such that sum of left array is equal to sum of right array
- Singleton design pattern in JavaScript
- Execute async functions in Series
- Convert entity relation array to object in Javascript
- Make high priority Api call in JavaScript
- Method chaining in JavaScript – Part 2
- Find all unique paths in a grid
- Create a toggle function in JavaScript

Prev Next

Comments

ปั๊มไลค์ says:

July 13, 2020 At 1:45 Pm

Like!! Really appreciate you sharing this blog post.Really thank you! Keep writing.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Start typing...

Name*

Name

Email*

Email

POST COMMENT

About Us Contact Us Privacy Policy Advertise

Handcrafted with ♥somewhere in **Mumbai**