Array.push() and unique items

Asked 3 years, 4 months ago Active 5 months ago Viewed 46k times



I have a simple case of pushing unique values into array. It looks like this:

```
32     this.items = [];

add(item) {
     if(this.items.indexOf(item) > -1) {
        this.items.push(item);
        console.log(this.items);
}
```

Seems pretty straight-forward, right? No, as it appears. It doesn't add any values. I am sure it's some kind of silly mistake on my side, but I can't seem to find it.

javascript

}

asked Apr 19 '16 at 13:10



Tomek Buszewski **3,386** 7 49 91

9 Answers



Yep its small mistake

58

```
if(this.items.indexOf(item) === -1) {
    this.items.push(item);
    console.log(this.items);
}
```

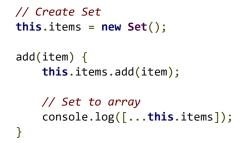


Ha ha, I don't know what I was thinking! Thanks! - Tomek Buszewski Apr 19 '16 at 13:15



You can use <u>Set</u> structure from ES6 to make your code faster and more readable:

20



answered Jun 11 '18 at 7:57



Sergey Zhigalov

only supported in chome - Gags Aug 29 '18 at 18:01

@Gags not true, work on all major recent browsers. developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/... – Bsienn Nov 16 '18 at 7:50

As a heads up. If like me you use JSON.stringify in your code, adding Sets will cause issues without further updates. - John Duskin Jan 24 at 16:48

I LOVE this. Had nearly forgotten about set. I'm going to be using this a lot, particularly nice in TypeScript. – Methodician Aug 27 at 20:55



I guess ES6 has set data structure, which you can use for unique entries





answered Apr 19 '16 at 14:30

Vivek Kumar

2,049 3 26 59

Set is also an array, store sequential data but only unique one not duplicate. - Vivek Kumar Apr 19 '16 at 16:52



Your logic is saying, "if this item exists already, then add it." It should be the opposite of that.

3 Change it to...



```
if (this.items.indexOf(item) == -1) {
    this.items.push(item);
}
```

answered Apr 19 '16 at 13:14





try .includes()



```
[1, 2, 3].includes(2);  // true
[1, 2, 3].includes(4);  // false
[1, 2, 3].includes(3, 3);  // false
[1, 2, 3].includes(3, -1);  // true
[1, 2, NaN].includes(NaN);  // true
```

so something like

```
const array = [1, 3];
if (!array.includes(2))
    array.push(2);
```

note the browser compatibility at the bottom of the page, however.

edited Apr 16 at 3:44

answered Apr 16 at 3:15 lachlan.p.jordan

lachlan.p.jord 85 2 7

1 I think this is definitely more elegant than .index0f . I never knew this existed before. - DarthCadeus Apr 16 at 4:01



You have to use == -1, if is equals to -1 doest not contains in your array

2

```
this.items = [];
add(item) {
   if(this.items.indexOf(item) == -1) {
      this.items.push(item);
      console.log(this.items);
   }
}
```

answered Apr 19 '16 at 13:14



fdfey

485 2



If you use **Lodash**, take a look _.union function:

1

```
let items = [];
items = _.union([item], items)
```

answered Aug 16 '18 at 11:45



1 441

3 16 34



so not sure if this answers your question but the indexOf the items you are adding keep returning -1. Not to familiar with js but it appears the items do that because they are not in the array yet. I made a jsfiddle of a little modified code for you.

0

this.items = [];

https://jsfiddle.net/jmpalmisano/Lnommkbw/2/

answered Apr 19 '16 at 13:21





```
var helper = {};
for(var i = 0; i < data.length; i++){
  helper[data[i]] = 1; // Fill object
}
var result = Object.keys(helper); // Unique items</pre>
```

answered Nov 29 '18 at 19:57

