



Interview Question

- How do you remove duplicates from an array?



Now we have a forum where you can discuss all the world's programming resources. [Read all about it.](#)



Diff Two Arrays

Compare two arrays and return a new array with any items only found in one of the two given arrays, but not both. In other words, return the symmetric difference of the two arrays.

Remember to use **Read-Search-Ask** if you get stuck. Try to pair program. Write your own code.

Here are some helpful links:

[Comparison Operators](#)

[Array.slice\(\)](#)

[Array.filter\(\)](#)

[Array.indexOf\(\)](#)

[Array.concat\(\)](#)

Run tests (ctrl + enter)

```
1 |
2 function diffArray(arr1, arr2) {
3   var newArr = [];
4   // Same, same; but different.
5   return newArr;
6 }
7
8 diffArray([1, 2, 3, 5], [1, 2, 3, 4, 5]);
9
```

A screenshot of a Safari browser window. The title bar shows "Safari File Edit View History". Below the title bar is a toolbar with standard Mac OS X icons for close, minimize, maximize, and back/forward navigation. The main content area has a green header bar with the text "freeCodeCamp(愈加)". Below this is a light blue box containing the text "Now we have a forum where you can...". The main content area contains several test cases for a function named "diffArray".

- ✓ `diffArray([1, 2, 3, 5], [1, 2, 3, 4, 5])` should return an array.
- ✗ `["diorite", "andesite", "grass", "dirt", "pink wool", "dead shrub"], ["diorite", "andesite", "grass", "dirt", "dead shrub"]` should return `["pink wool"]`.
- ✗ `["andesite", "grass", "dirt", "pink wool", "dead shrub"], ["diorite", "andesite", "grass", "dirt", "dead shrub"]` should return `["diorite", "pink wool"]`.
- ✓ `["andesite", "grass", "dirt", "dead shrub"], ["andesite", "grass", "dirt", "dead shrub"]` should return `[]`.
- ✗ `[1, 2, 3, 5], [1, 2, 3, 4, 5]` should return `[4]`.

Return What's Different

Here are some helpful links:

[**Comparison Operators**](#)

[**Array.slice\(\)**](#)

[**Array.filter\(\)**](#)

[**Array.indexOf\(\)**](#)

[**Array.concat\(\)**](#)

Array.slice() method

Examples

Return a portion of an existing array

```
1 | var fruits = ['Banana', 'Orange', 'Lemon', 'Apple', 'Mango'];
2 | var citrus = fruits.slice(1, 3);
3 |
4 | // citrus contains ['Orange', 'Lemon']
```

Array.indexOf() method

Syntax

```
string.indexOf(searchvalue,start)
```

Parameter Values

| Parameter | Description |
|-------------|--|
| searchvalue | Required. The string to search for |
| start | Optional. Default 0. At which position to start the search |

Technical Details

Return Value: A Number, representing the position where the specified searchvalue occurs for the first time, or -1 if it never occurs

JavaScript Version: 1.0

```
var str = "Hello world, welcome to the universe.";
var n = str.indexOf("e");
```

The result of *n* will be:

```
1
```

Array.concat() method

Join two arrays:

```
var hege = ["Cecilie", "Lone"];
var stale = ["Emil", "Tobias", "Linus"];
var children = hege.concat(stale);
```

The values of the *children* array will be:

```
Cecilie, Lone, Emil, Tobias, Linus
```

Array.filter() method

```
var testArray = [
    {name:"John", age: 40, children: 2,
     company: { name: "MegaCorp", employees: 200}},
    {name:"Sue",   age: 30, children: 1,
     company:{ name: "MegaCorp", employees: 200}},
    {name:"Mary",  age: 55, children: 3,
     company:{ name: "MegaCorp", employees: 200}},
    {name:"Jack",  age: 20, children: 0,
     company:{ name: "MiniCorp", employees: 100}}];

// Using a for loop
var filtered = [];
for (var k = 0; k < testArray.length; ++k) {
    var item = testArray[k];
    if (item.age == 40 && item.age == 30) filtered.push(item);
}

// Using Array.filter
testArray.filter(function(item) {
    return item.age == 40 || item.age == 30
}); // returns John
```

How can we solve?

- We can concat the two parts of diffArray and filter
- We can use a map or loop to filter through each one and compare

```
1
2 function diffArray(arr1, arr2) {
3
4   var newArr = arr1.concat(arr2); //instead of looping through two different arr1 and arr2
5   //we can combine arr1 and arr2 with the concat method
6   return newArr.filter(function(item) {
7     return !(arr1.indexOf(item) >-1 && arr2.indexOf(item) >-1);
8     //indexOf is useful because -1 shows unique elements
9     //the filter method allows us to filter through each element in an array
10    });
11  // Same, same; but different.
12
13
14 }
15 diffArray([1, 2, 3, 5], [1, 2, 3, 4, 5]);
16
```

Additional Reference

- <http://stackoverflow.com/questions/15004759/how-to-filter-an-array-with-index-of-condition-in-javascript>
- <http://jszen.com/best-way-to-get-unique-values-of-an-array-in-javascript.7.html--shows>
- <http://stackoverflow.com/questions/143847/best-way-to-find-if-an-item-is-in-a-javascript-array>