forEach / map / filter

Write the functions necessary for the tests to pass, you can find the tests in the downloaded code.

js-array-foreach-map-filter-exercise-starter-code.zip 3.5KB

forEach

doubleValues

Write a function called doubleValues which accepts an array and returns a new array with all the values in the array passed to the function doubled

```
doubleValues([1,2,3]) // [2,4,6] doubleValues([5,1,2,3,10]) // [10,2,4,6,20]
```

onlyEvenValues

Write a function called onlyEvenValues which accepts an array and returns a new array with only the even values in the array passed to the function

```
onlyEvenValues([1,2,3]) // [2] onlyEvenValues([5,1,2,3,10]) // [2,10]
```

showFirstAndLast

Write a function called showFirstAndLast which accepts an array of strings and returns a new array with only the first and last character of each string.

```
showFirstAndLast(['colt','matt', 'tim', 'test']) // ["ct", "mt", "tm", "tt"] showFirstAn
dLast(['hi', 'goodbye', 'smile']) // ['hi', 'ge', 'se']
```

addKeyAndValue

Write a function called addKeyAndValue which accepts an array of objects, a key, and a value and returns the array passed to the function with the new key and value added for each object

```
addKeyAndValue( [ {name: 'Elie'}, {name: 'Tim'}, {name: 'Matt'}, {name: 'Colt'} ], 'titl
e', 'instructor' ) /* [ {name: 'Elie', title:'instructor'}, {name: 'Tim', title:'instruc
tor'}, {name: 'Matt', title:'instructor'}, {name: 'Colt', title:'instructor'} ] */
```

vowelCount

Write a function called vowelCount which accepts a string and returns an object with the keys as the vowel and the values as the number of times the vowel appears in the string. This function should be case insensitive so a lowercase letter and uppercase letter should count

```
vowelCount('Elie') // {e:2,i:1}; vowelCount('Tim') // {i:1}; vowelCount('Matt') // {a:
1}) vowelCount('hmmm') // {}; vowelCount('I Am awesome and so are you') // {i: 1, a: 4,
e: 3, o: 3, u: 1}
```

map

doubleValuesWithMap

Write a function called doubleValuesWithMap which accepts an array and returns a new array with all the values in the array passed to the function doubled

```
doubleValuesWithMap([1,2,3]) // [2,4,6] doubleValuesWithMap([1,-2,-3]) // [2,-4,-6]
```

function doubleValuesWithMap(arr) {}

valTimesIndex

Write a function called valTimesIndex which accepts an array and returns a new array with each value multiplied by the index it is currently at in the array.

```
valTimesIndex([1,2,3]) // [0,2,6] valTimesIndex([1,-2,-3]) // [0,-2,-6]
```

extractKey

Write a function called extractKey which accepts an array of objects and some key and returns a new array with the value of that key in each object.

```
extractKey( [ {name: 'Elie'}, {name: 'Tim'}, {name: 'Matt'}, {name: 'Colt'} ], 'name' )
// ['Elie', 'Tim', Matt', 'Colt']
```

extractFullName

Write a function called extractFullName which accepts an array of objects and returns a new array with the value of the key with a name of “first” and the value of a key with the name of “last” in each object, concatenated together with a space.

```
/* extractFullName([ {first: 'Elie', last:"Schoppik"}, {first: 'Tim', last:"Garcia"}, {f
irst: 'Matt', last:"Lane"}, {first: 'Colt', last:"Steele"} ]) // ['Elie Schoppik', 'Tim
Garcia', 'Matt Lane', 'Colt Steele'] */
```

filter

filterByValue

Write a function called filterByValue which accepts an array of objects and a key and returns a new array with all the objects that contain that key.

```
filterByValue( [ {first: 'Elie', last:"Schoppik"}, {first: 'Tim', last:"Garcia", isCatOwner: true}, {first: 'Matt', last:"Lane"}, {first: 'Colt', last:"Steele", isCatOwner: true} ], 'isCatOwner' ) /* [ {first: 'Tim', last:"Garcia", isCatOwner: true}, {first: 'Colt', last:"Steele", isCatOwner: true} ] */
```

find

Write a function called find which accepts an array and a value and returns the first element in the array that has the same value as the second parameter or undefined if the value is not found in the array.

```
find([1,2,3,4,5], 3) // 3 find([1,2,3,4,5], 10) // undefined
```

findInObj

Write a function called findInObj which accepts an array of objects, a key, and some value to search for and returns the first found value in the array.

```
findInObj( [ {first: 'Elie', last:"Schoppik"}, {first: 'Tim', last:"Garcia", isCatOwner: true}, {first: 'att', last:"Lane"}, {first: 'Colt', last:"Steele", isCatOwner: true} ], 'isCatOwner', true ) // {first: 'Tim', last:"Garcia", isCatOwner: true}
```

removeVowels

Write a function called removeVowels which accepts a string and returns a new string with all of the vowels (both uppercased and lowercased) removed. Every character in the new string should be lowercased.

```
removeVowels('Elie') // ('l') removeVowels('TIM') // ('tm') removeVowels('ZZZZZZ') //
('zzzzzz')
```

doubleOddNumbers

Write a function called doubleOddNumbers which accepts an array and returns a new array with all of the odd numbers doubled (HINT - you can use map and fitler to double and then filter the odd numbers).

JavaScript Copy

```
doubleOddNumbers([1,2,3,4,5]) // [2,6,10] doubleOddNumbers([4,4,4,4,4]) // []
```

Solution