
# Reduce

js-array-reduce-exercise-starter.zip 2.3KB

## extractValue

Write a function called extractValue which accepts an array of objects and a key and returns a new array with the value of each object at the key.

```
const arr = [{name: 'Elie'}, {name: 'Tim'}, {name: 'Matt'}, {name: 'Colt'}] extractValue
(arr,'name') // ['Elie', 'Tim', 'Matt', 'Colt']
```

## 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};
```

## addKeyAndValue

Write a function called addKeyAndValue which accepts an array of objects and returns the array of objects passed to it with each object now including the key and value passed to the function.

```
const arr = [{name: 'Elie'}, {name: 'Tim'}, {name: 'Matt'}, {name: 'Colt'}]; addKeyAndVa
lue(arr, 'title', 'Instructor') // [ {title: 'Instructor', name: 'Elie'}, {title: 'Instr
uctor', name: 'Tim'}, {title: 'Instructor', name: 'Matt'}, {title: 'Instructor', name:
'Colt'} ]
```

## partition

Write a function called partition which accepts an array and a callback and returns an array with two arrays inside of it. The partition function should run the callback function on each value in the array and if the result of the callback function at that specific value is true, the value should be placed in the first subarray. If the result of the callback function at that specific value is false, the value should be placed in the second subarray.

```
function isEven(val){ return val % 2 === 0; } const arr = [1,2,3,4,5,6,7,8]; partition(arr, isEven
8], [1,3,5,7]]; function isLongerThanThreeCharacters(val){ return val.length > 3; } const names =
t', 'Tim', 'Matt']; partition(names, isLongerThanThreeCharacters) // [['Elie', 'Colt', 'Matt'], ['
```

## Solution

```
function extractValue(arr, key){ return arr.reduce(function(acc,next){ acc.push(next
[key]); return acc; },[]); } function vowelCount(str){ const vowels = "aeiou"; retur
n str.split('').reduce(function(acc,next){ let lowerCased = next.toLowerCase() if(vo
wels.indexOf(lowerCased) !== -1){ if(acc[lowerCased]){ acc[lowerCased]++; } else { a
cc[lowerCased] = 1; } } return acc; }, {}); } function addKeyAndValue(arr, key, valu
e){ return arr.reduce(function(acc,next,idx){ acc[idx][key] = value; return acc; },a
rr); } function partition(arr, cb){ return arr.reduce(function(acc,next){ if(cb(nex
t)){ acc[0].push(next); } else { acc[1].push(next); } return acc; }, [[],[]]); }
```