**How to Convert a Map to an Array in JavaScript -** [Coding Beauty](https://codingbeauty.medium.com/?source=post_page-----9c93fe20e647--------------------------------) May 28, 2022

Learn multiple ways to easily convert a Map object to an array of objects in JavaScript.

In this article, we’ll be learning different ways to convert a Map object to an array of objects in JavaScript.

**1. Array from() Method**

To convert a Map to an array, we can use the static Array.from() method, passing the Map as the first argument, and a map function to transform the Map entries as a second argument.

const map = new Map();  
map.set('user1', 'John');  
map.set('user2', 'Kate');  
map.set('user3', 'Peter');const arr = Array.from(map, function (entry) {  
 return { key: entry[0], value: entry[1] };  
});/\*\*  
 [  
 { key: 'user1', value: 'John' },  
 { key: 'user2', value: 'Kate' },  
 { key: 'user3', value: 'Peter' }  
 ]  
\*/  
console.log(arr);

**Note**: We can use array destructuring and an arrow function to shorten the code above:

const arr = Array.from(map, ([key, value]) => ({  
 key,  
 value,  
}));

**Note**: To convert the Map to an array of key-value objects:

const arr = Array.from(map, ([key, value]) => ({  
 [key]: value,  
}));// [ { user1: 'John' }, { user2: 'Kate' }, { user3: 'Peter' } ]  
console.log(arr);

**Note**: If we don’t pass the map function as an argument, Array.from() will return an array of key-value pairs, one pair for each Map entry. For example:

const map = new Map();  
map.set('user1', 'John');  
map.set('user2', 'Kate');  
map.set('user3', 'Peter');const arr = Array.from(map);// [ [ 'user1', 'John' ], [ 'user2', 'Kate' ], [ 'user3', 'Peter' ] ]  
console.log(arr);

This means we can use the Array map() method to transform this array into an array of objects:

const map = new Map();  
map.set('user1', 'John');  
map.set('user2', 'Kate');  
map.set('user3', 'Peter');const arr = Array.from(map).map(([key, value]) => ({  
 key,  
 value,  
}));/\*\*  
 [  
 { key: 'user1', value: 'John' },  
 { key: 'user2', value: 'Kate' },  
 { key: 'user3', value: 'Peter' }  
 ]  
\*/  
console.log(arr);

**2. Spread Operator and Array map()**

Using the JavaScript spread operator (...) on a Map will unpack its entries into an array of key-value pairs, just like Array.from() does when it doesn't receive a callback. For example:

const map = new Map();  
map.set('user1', 'John');  
map.set('user2', 'Kate');  
map.set('user3', 'Peter');const arr = [...map];// [ [ 'user1', 'John' ], [ 'user2', 'Kate' ], [ 'user3', 'Peter' ] ]  
console.log(arr);

This means we can also convert a Map to an array of objects by combining the spread operator with the Array map() method:

const map = new Map();  
map.set('user1', 'John');  
map.set('user2', 'Kate');  
map.set('user3', 'Peter');const arr = [...map].map(([key, value]) => ({  
 key,  
 value,  
}));/\*\*  
 [  
 { key: 'user1', value: 'John' },  
 { key: 'user2', value: 'Kate' },  
 { key: 'user3', value: 'Peter' }  
 ]  
\*/  
console.log(arr);

**3. Iterating over Map and Adding Elements to Array**

Another way to convert a Map to an array is to iterate over each Map entry, create an object with the entry and add the object to the resulting array. For example, we can iterate over the Map with the Map forEach() method, and add array elements with the Array push() method.

const map = new Map();  
map.set('user1', 'John');  
map.set('user2', 'Kate');  
map.set('user3', 'Peter');const arr = [];  
map.forEach((value, key) => arr.push({ key, value }));/\*\*  
 [  
 { key: 'user1', value: 'John' },  
 { key: 'user2', value: 'Kate' },  
 { key: 'user3', value: 'Peter' }  
 ]  
\*/  
console.log(arr);

Alternatively, we can use the for...of loop to iterate over the Map:

const map = new Map();  
map.set('user1', 'John');  
map.set('user2', 'Kate');  
map.set('user3', 'Peter');const arr = [];  
for (const [key, value] of map) {  
 arr.push({ key, value });  
}/\*\*  
 [  
 { key: 'user1', value: 'John' },  
 { key: 'user2', value: 'Kate' },  
 { key: 'user3', value: 'Peter' }  
 ]  
\*/  
console.log(arr);