**Looping Through Maps**

You’ve created a Map, added some key-value pairs, and now you want to loop through your Map. Thankfully, you’ve got three different options to choose from:

1. Step through each key or value using the Map’s default iterator
2. Loop through each key-value pair using the new for...of loop
3. Loop through each key-value pair using the Map’s .forEach() method

**1. Using the MapIterator**

Using both the .keys() and .values() methods on a Map will return a new iterator object called MapIterator. You can store that iterator object in a new variable and use .next() to loop through each key or value. Depending on which method you use, will determine if your iterator has access to the Map’s keys or the Map’s values.

**let** iteratorObjForKeys = members.keys();

iteratorObjForKeys.next();

*Object {value: 'Evelyn', done: false}*

Use .next() to the get the next key value.

iteratorObjForKeys.next();

*Object {value: 'Liam', done: false}*

And so on.

iteratorObjForKeys.next();

*Object {value: 'Sophia', done: false}*

On the flipside, use the .values() method to access the Map’s values, and then repeat the same process.

**let** iteratorObjForValues = members.values();

iteratorObjForValues.next();

*Object {value: 75.68, done: false}*

**2. Using a for...of Loop**

Your second option for looping through a Map is with a for...of loop.

**for** (**const** member **of** members) {

console.log(member);

}

*['Evelyn', 75.68]*

*['Liam', 20.16]*

*['Sophia', 0]*

*['Marcus', 10.25]*

However, when you use a for...of loop with a Map, you don’t exactly get back a key or a value. Instead, the key-value pair is split up into an array where the first element is the key and the second element is the value. If only there were a way to fix this?

/\*

\* Using array destructuring, fix the following code to print the keys and values of the `members` Map to the console.

\*/

const members = new Map();

members.set('Evelyn', 75.68);

members.set('Liam', 20.16);

members.set('Sophia', 0);

members.set('Marcus', 10.25);

let a,b;

for (const member of members) {

[a,b] = member;

console.log(a , b);

}

## 3. Using a forEach Loop

Your last option for looping through a Map is with the .forEach() method.

members.forEach((value, key) => console.log(key, value));

*'Evelyn' 75.68*

*'Liam' 20.16*

*'Sophia' 0*

*'Marcus' 10.25*

Notice how with the help of an arrow function, the forEach loop reads fairly straightforward. For each value and key in members, log the value and key to the console.