The last step to working with Sets is looping over them.

If you remember back to our discussion on the new *iterable* and *iterator protocols* in ES6, then you’ll recall that Sets are built-in iterables. This means two things in terms of looping:

1. You can use the Set’s default iterator to step through each item in a Set, one by one.
2. You can use the new for...of loop to loop through each item in a Set.

**Using the SetIterator**

Because the .values() method returns a new iterator object (called SetIterator), you can store that iterator object in a variable and loop through each item in the Set using .next().

**const** iterator = months.values();

iterator.next();

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

And if you run .next() again?

iterator.next();

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

And so on until done equals true which marks the end of the Set.

Replay

Mute

Loaded: 100.00%

Remaining Time -0:00

1x

Playback Rate

Picture-in-Picture

Fullscreen

**Using a for...of Loop**

An easier method to loop through the items in a Set is the for...of loop.

**const** colors = **new** Set(['red', 'orange', 'yellow', 'green', 'blue', 'violet', 'brown', 'black']);

**for** (**const** color **of** colors) {

console.log(color);

}

*red  
orange  
yellow  
green  
blue  
violet  
brown  
black*