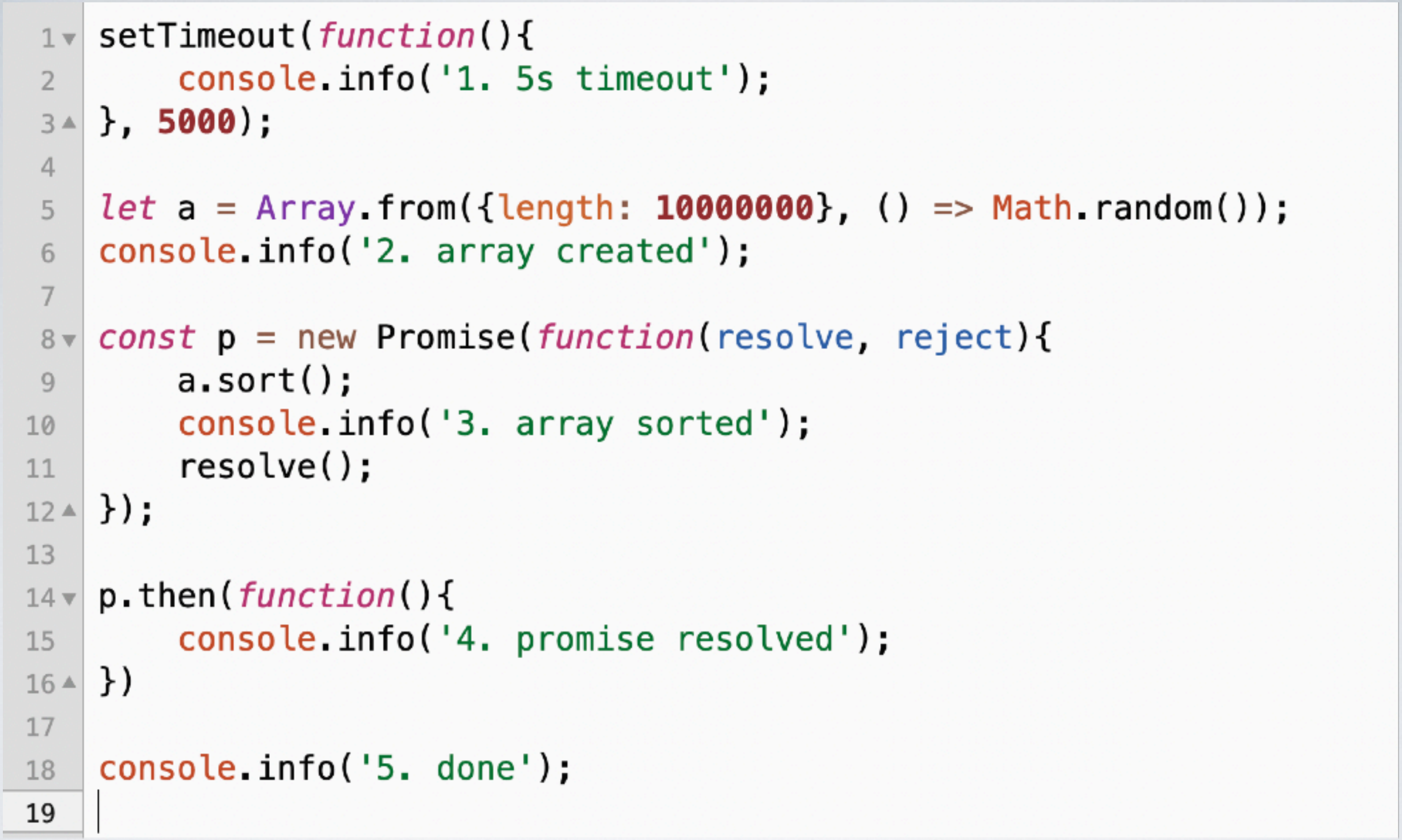

```javascript
setTimeout(function(){
    console.info('1. 5s timeout');
}, 5000);

let a = Array.from({length: 10000000}, () => Math.random());
console.info('2. array created');

const p = new Promise(function(resolve, reject){
    a.sort();
    console.info('3. array sorted');
    resolve();
});

p.then(function(){
    console.info('4. promise resolved');
})

console.info('5. done');
```

- ➡ Prints 2, 3, 5, 4, 1

Calling a promise with async/await

```
async function run() {
     const data = await readFile(filepath);
     console.log(data);
};

run().catch(err => console.error(err));
```

✓ `Async` returns a promise

◉ `Await` suspends the computation and waits for the promise to resolve

➡ The remaining code after the `await` expression is placed into the **microtask queue**