

Microtasks vs Macrotasks in the Event Loop

- **Microtasks** (like Promise `.then()` callbacks)
have higher priority than macrotasks
- **Macrotasks**
(like `setTimeout()`, I/O operations, and DOM events)
are processed only after the microtask queue is empty

```
1▼ setTimeout(function(){
2    console.info('1. 5s timeout');
3▲ }, 5000);
4
5 let a = Array.from({length: 10000000}, () => Math.random());
6 console.info('2. array created');
7
8▼ const p = new Promise(function(resolve, reject){
9    a.sort();
10    console.info('3. array sorted');
11    resolve();
12▲ });
13
14▼ p.then(function(){
15    console.info('4. promise resolved');
16▲ })
17
18 console.info('5. done');
19 |
```

➡ Prints 2, 3, 5, 4, 1