

## Event Loop

### 1-) Meditation Countdown

Build a UI that contains the following elements:

- One input element to represent starting number of seconds
- Two buttons: Start, Stop

Your Meditation Countdown will work as following:

- Initially, the input is set to 0, "Start" and "Stop" buttons are disabled.
- When users change the input value of the timer to any value other than 0, then activate the "Start" button.
- Users will add the number of seconds in the input element and click "Start".
- "Start" button will be disabled, "Stop" button will be enabled, and the countdown timer will decrease.
- Reset the timer to "0" if users click "Stop".
- When we naturally reach 0 then we should disable "Start", stop the timer, and deactivate the "Stop" button.

### 2-) Change isPrime() that takes in a single number parameter and returns a new promise.

- Using setTimeout and after 500 milliseconds, the promise will either resolve or reject.
- If the input is a prime number, the promise resolves with {prime: true}.
- If the input is not a prime number, it rejects with {prime: false}.

Write code and call your promisified function and test it for both scenarios (resolve and reject)

3-)Remember removeDuplicates() method that you wrote earlier? Currently this method runs synchronously. Rewrite an asynchronous version of it removeDuplicatedAsync() as following:

```
console.log(`start`);
[4, 1, 5, 7, 2, 3, 1, 4, 6, 5, 2].removeDuplicatedAsync();
console.log(`end`);

// Output:
// start
// end
// [4, 1, 5, 7, 2, 3, 6]
```

<https://classroom.github.com/a/4ohLb8Fe>