

```

    return Math.floor(Math.random() * max);
}

function createPromiseWithTimeout() {
    return new Promise(function executor(resolve, reject) {
        console.log("Entering the executor callback in the promise constructor");
        setTimeout(function () {
            let num = getRandomInt(10);
            if(num % 2 == 0) {
                // if the random number is even we fullfill
                resolve(num);
            } else {
                // if the random number is odd we reject
                reject(num);
            }
        }, 10000);
        console.log("Exiting the executor callback in the promise constructor");
    });
}

console.log("Starting...");
const p = createPromiseWithTimeout();
console.log("We are now waiting for the promise to complete");
console.log("Currently my promise object is like ... ", p);
p
    .then(
        function fulfillHandler(value) {
            console.log("inside fulfill handler with value", value);
            console.log("Promise after fulfillment is", p);
        },
        function rejectionHandler(value) {
            console.log("Inside rejection handler with value", value);
            console.log("Promise after rejection is", p);
        }
    );

```

2. Under 6
 6.1 → 0-fulfill
 only get: [x, y]

count
 1000

Runtu
 timer 1 → 10s

→ callback queue

microtask / job queue