# BENDING TIME WITH RXJS

#FULLSTACKCONF**2019**

@_MAXGALLO

HI 👋

# I'M MAX GALLO

PRINCIPAL ENGINEER **@ DAZN**

TWITTER: **@_MAXGALLO**

# AGENDA

1. USING SCHEDULERS
2. MICROTASKS AND MACROTASKS
3. BENDING TIME

```
const { of } = require('rxjs');

const observer = {
    next(value) {
        console.log(value);
    }
}

of(1, 2, 3, 4)
    .subscribe(observer);

console.log('subscribe');
```

USING **NO SCHEDULER** THE OUTPUT IS **1, 2, 3, 4, SUBSCRIBE**

```
const { of, scheduled, asapScheduler } = require('rxjs');

const observer = {
    next(value) {
        console.log(value);
    },
}

scheduled(of(1, 2, 3, 4), asapScheduler)
  .subscribe(observer);

console.log('subscribe');
```

USING **ASAP SCHEDULER** THE OUTPUT IS **SUBSCRIBE, 1, 2, 3, 4**

```
const { of, scheduled, asyncScheduler } = require('rxjs');

const observer = {
    next(value) {
        console.log(value);
    },
}

scheduled(of(1, 2, 3, 4), asyncScheduler)
  .subscribe(observer);

console.log('subscribe');
```

USING **ASYNC SCHEDULER** THE OUTPUT IS **SUBSCRIBE, 1, 2, 3, 4**

# RECAP 🤔

| | |
|---|---|
| NO SCHEDULER | 1, 2, 3, 4, SUBSCRIBE |
| ASAP SCHEDULER | SUBSCRIBE, 1, 2, 3, 4 |
| ASYNC SCHEDULER | SUBSCRIBE, 1, 2, 3, 4 |

## LETS RACE THEM 🏎💨

```
const {
    of, merge, scheduled,
    asapScheduler,
    asyncScheduler,
    queueScheduler,
} = require('rxjs');

const observer = {
    next(value) {
        console.log(value);
    },
}

const asapObs = scheduled(of('ASAP'), asapScheduler)
const asyncObs = scheduled(of('ASYNC'), asyncScheduler)
const queueObs = scheduled(of('QUEUE'), queueScheduler)

merge(asapObs, asyncObs, queueObs)
  .subscribe(observer);

console.log('SUBSCRIBE');
```

THE OUTPUT IS **QUEUE, SUBSCRIBE, ASAP, ASYNC**

## SCHEDULER TYPES

| TYPE | EXECUTION |
|---|---|
| QUEUE | SYNC |
| ASAP | ASYNC → MICROTASK |
| ASYNC | ASYNC → MACROTASK |

# DEEP DIVE **MACROTASKS** & **MICROTASKS**

## TWO QUEUES OF THE EVENT LOOP

```
MACROtasks      MICROtasks
----------      ----------
setTimeout      process.nextTick
setInterval     Promises
setImmediate    MutationObserver
I/O tasks
```

## DEEP DIVE **MACROTASKS** & **MICROTASKS**

**FOR EVERY MACROTASK REMOVED FROM THE MACRO-QUEUE, ALL THE MICROTASKS ARE REMOVED FROM THE MICRO-QUEUE.**

(HOW TO REMEMBER: 👩‍✈️ 🛂 ✈️)

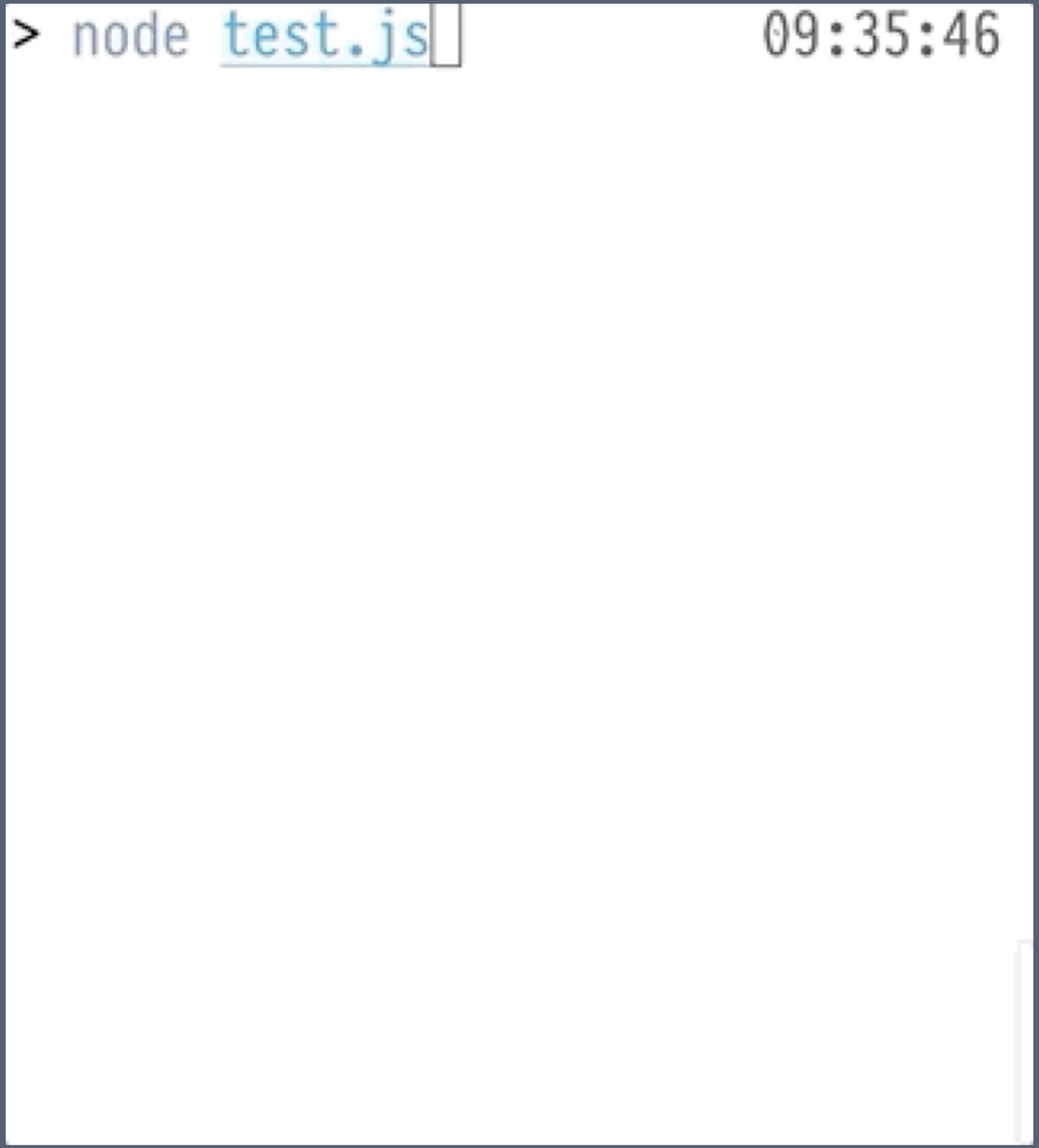
TO LEARN MORE WHATCH HTTPS://WWW.YOUTUBE.COM/WATCH?V=U1KQX6AENYW AND READ HTTPS://JAKEARCHIBALD.COM/2015/TASKS-MICROTASKS-QUEUES-AND-SCHEDULES/

## SCHEDULERS CONTROL THE ORDER OF EVENT EMISSION

```
const { interval } = require('rxjs');
const { take } = require('rxjs/operators');

const observer = {
    next(value) {
        console.log(value);
    }
}

interval(1000)
    .pipe(take(3600))
    .subscribe(observer);
```



```
> node test.js                         09:35:46
```

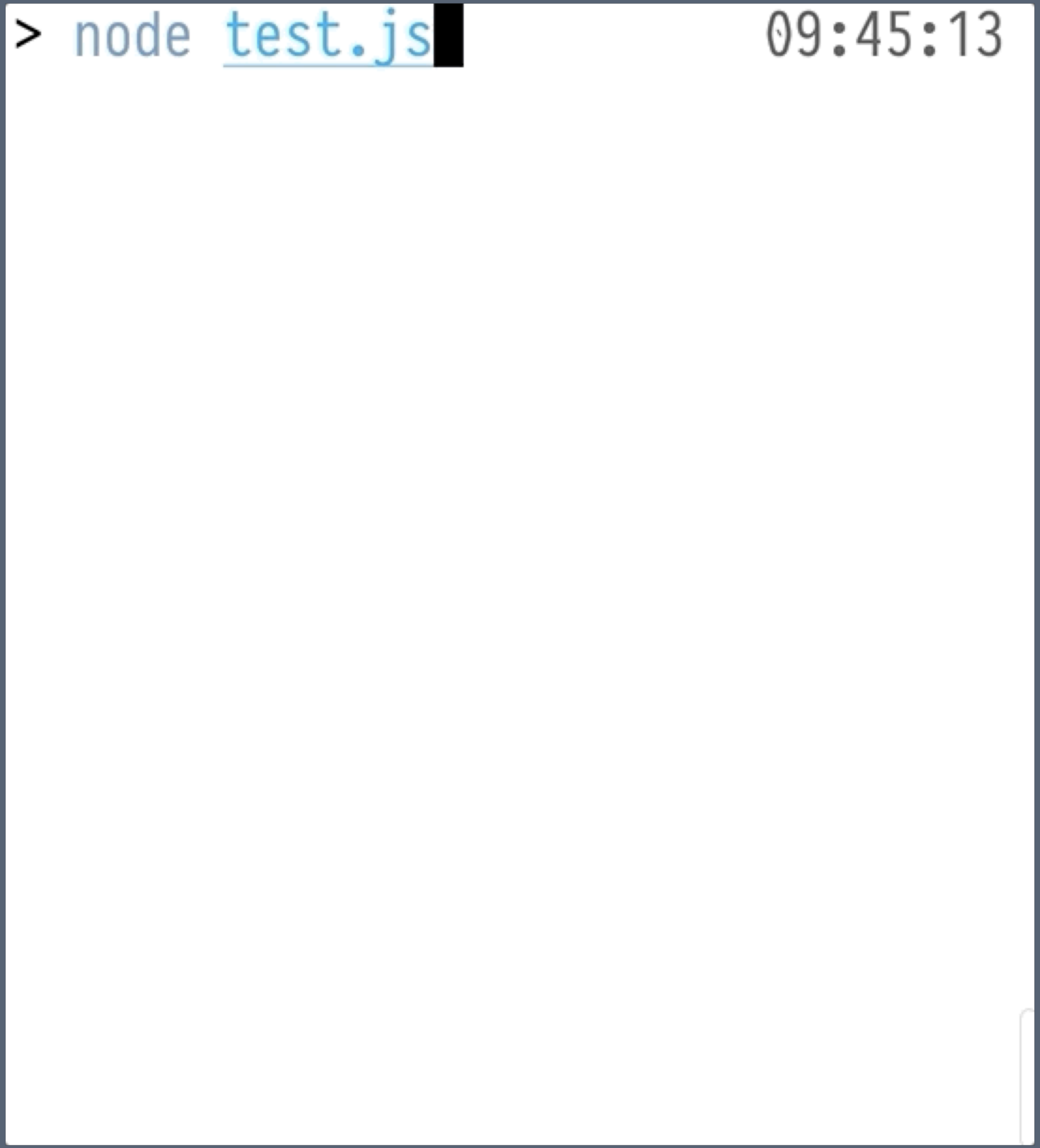
```js
const {
    interval,
    VirtualTimeScheduler
} = require('rxjs');
const { take } = require('rxjs/operators');

const scheduler = new VirtualTimeScheduler();

const observer = {
    next(value) {
        console.log(value);
    }
}

interval(1000, scheduler)
    .pipe(take(3600))
    .subscribe(observer);

scheduler.flush();
```



```
> node test.js                    09:45:13
```

## SCHEDULERS CONTROL THE SPEED OF EVENT EMISSION

## TAKEAWAYS 📝

## 1. EVENT LOOP HAS TWO QUEUES, MACROTASKS AND MICROTASKS

## 2. SCHEDULERS **CONTROL THE** **ORDER** **AND THE** **SPEED** **OF EVENT EMISSION**

# THANK YOU 🙏

[GITHUB.COM/MAXGALLO/TALK-BENDING-TIME-WITH-RXJS](https://github.com/maxgallo/talk-bending-time-with-rxjs)

**@_MAXGALLO**