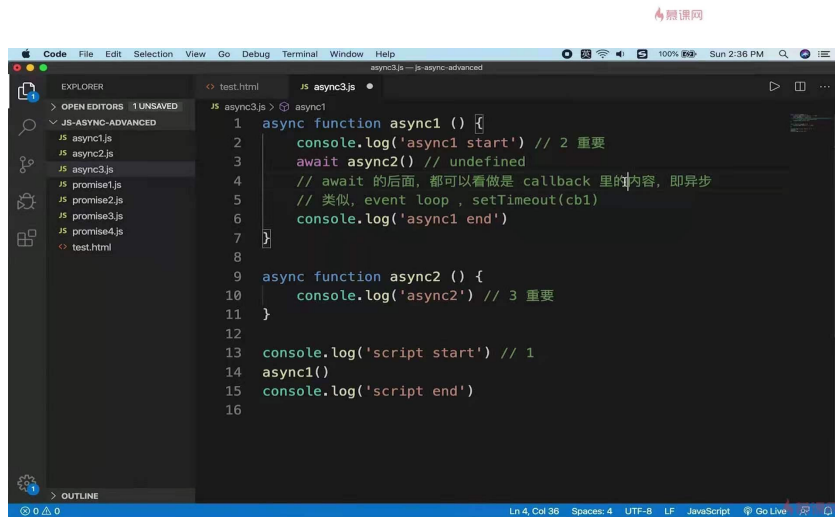
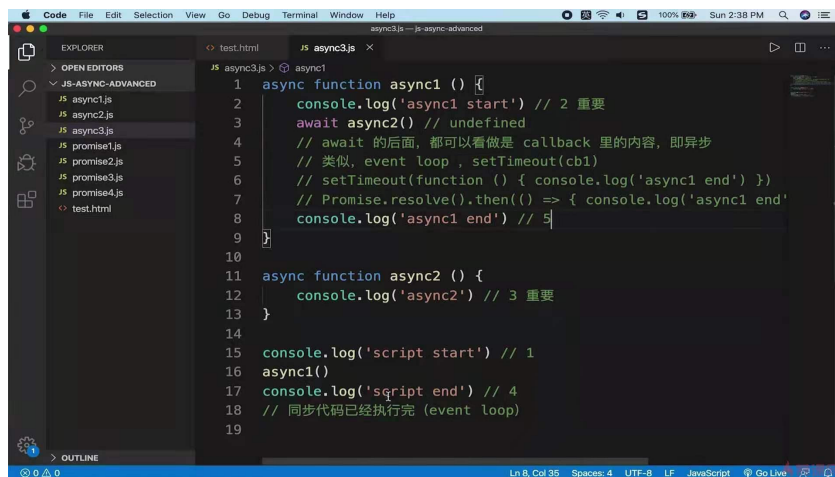


异步的本质

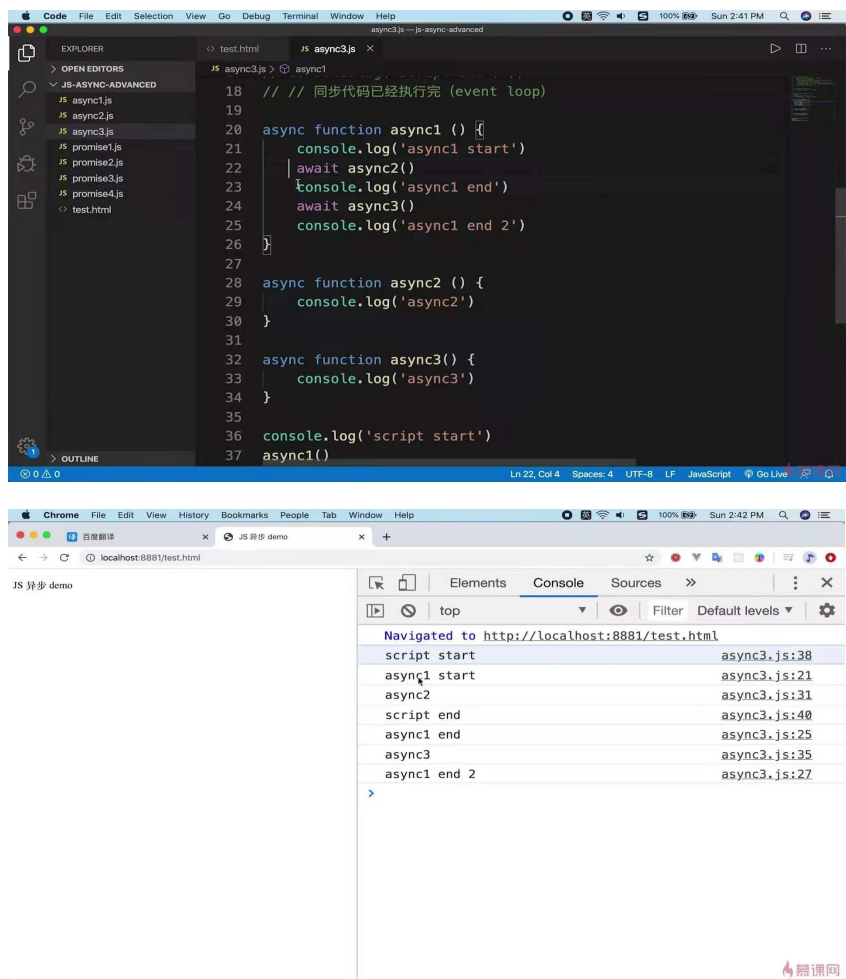
- ◆ async/await 是消灭异步回调的终极武器
- ◆ JS 还是单线程，还得是有异步，还得是基于 event loop
- ◆ async/await 只是一个语法糖，但这颗糖真香！



```
1 async function async1 () {
2   console.log('async1 start') // 2 重要
3   await async2() // undefined
4   // await 的后面，都可以看做是 callback 里的内容，即异步
5   // 类似，event loop , setTimeout(cb1)
6   console.log('async1 end')
7 }
8
9 async function async2 () {
10  console.log('async2') // 3 重要
11 }
12
13 console.log('script start') // 1
14 async1()
15 console.log('script end')
16
```



```
1 async function async1 () {
2   console.log('async1 start') // 2 重要
3   await async2() // undefined
4   // await 的后面，都可以看做是 callback 里的内容，即异步
5   // 类似，event loop , setTimeout(cb1)
6   // setTimeout(function () { console.log('async1 end') })
7   // Promise.resolve().then(() => { console.log('async1 end')
8   console.log('async1 end') // 5
9 }
10
11 async function async2 () {
12  console.log('async2') // 3 重要
13 }
14
15 console.log('script start') // 1
16 async1()
17 console.log('script end') // 4
18 // 同步代码已经执行完 (event loop)
19
```



for ... of

- ◆ for ... in (以及 forEach for) 是常规的同步遍历
- ◆ for ... of 常用于异步的遍历

async/await 总结

- ◆ async/await 解决了异步回调，是一个很香的语法糖
- ◆ async/await 和 Promise 的关系，重要！
- ◆ for...of 的使用