

本章主要内容

- ◆ event loop
- ◆ 微任务/宏任务
- ◆ promise 进阶
- ◆ async/await

www.imoooc.com 慕课网

问答题

- ◆ 请描述 event loop (事件循环/事件轮询) 的机制, 可画图
- ◆ 什么是宏任务和微任务, 两者有什么区别?
- ◆ Promise 有哪三种状态? 如何变化?

慕课网

场景题 - promise then 和 catch 的连接

```
// 第一题
Promise.resolve().then(() => {
  console.log(1)
}).catch(() => {
  console.log(2)
}).then(() => {
  console.log(3)
})

// 第二题
Promise.resolve().then(() => {
  console.log(1)
  throw new Error('error')
}).catch(() => {
  console.log(2)
}).then(() => {
  console.log(3)
})

// 第三题
Promise.resolve().then(() => {
  console.log(1)
  throw new Error('error')
}).catch(() => {
  console.log(2)
}).catch(() => { // 这里是 catch
  console.log(3)
})
```

慕课网

场景题 - async/await 语法

```
async function fn() {
  return 100
}

(async function () {
  const a = fn() // ??
  const b = await fn() // ??
})();

(async function () {
  console.log('start')
  const a = await 100
  console.log('a', a)
  const b = await Promise.resolve(200)
  console.log('b', b)
  const c = await Promise.reject(300)
  console.log('c', c)
  console.log('end')
})(); // 执行完毕, 打印出那些内容?
```

慕课网

场景题 - promise 和 setTimeout 的顺序

```
console.log(100)
setTimeout(() => {
  console.log(200)
})
Promise.resolve().then(() => {
  console.log(300)
})
console.log(400)
```

慕课网

场景题 - 外加 async/await 的顺序问题

```
async function async1 () {
  console.log('async1 start')
  await async2()
  console.log('async1 end')
}
async function async2 () {
  console.log('async2')
}
console.log('script start')

setTimeout(function () {
  console.log('setTimeout')
}, 0)
```

```
// 连接左侧代码，一起阅读
async1()

new Promise (function (resolve) {
  console.log('promise1')
  resolve()
}).then (function () {
  console.log('promise2')
})

console.log('script end')
```

慕课网

inoooc

event loop (事件循环/事件轮询)

- ◆ JS 是单线程运行的
- ◆ 异步要基于回调来实现
- ◆ event loop 就是异步回调的实现原理

慕课网

JS 如何执行？

- ◆ 从前到后，一行一行执行
- ◆ 如果某一行执行报错，则停止下面代码的执行
- ◆ 先把同步代码执行完，再执行异步

示例

```
console.log('Hi')

setTimeout(function cb1() {
  console.log('cb1') // cb 即 callback
}, 5000)

console.log('Bye')
```

开始讲解 event loop 过程

- ◆ 初学者可能会感觉难，尽量深入浅出
- ◆ 第一遍讲解时，感觉不懂的不要停下，会重复讲三遍
- ◆ 不要扣细节，不要扩大范围，核心是 event loop 的过程