

通用异步编程概念

1. 阻塞
2. 线程 => JS 单线程
 - a. CPU有多个内核 (core) 、多线程
 - b. 主线程 (main thread)
3. 异步代码
 - a. Web Workers
 - b. Promise

异步JavaScript简介

1. 同步js 按顺序执行
2. 异步js
 - a. callbacks - old school
 - b. promise - new school
3. 异步callbacks
 - a. [addEventListener\(\)](#)
 - b. [XMLHttpRequest API](#)
4. Promises
 - a. 现代版的, 更高效的 [XMLHttpRequest](#)
 - i. .then()
 - ii. .catch()
5. Promises 对比 callbacks
 - a. promises与旧式callbacks有一些相似之处。它们本质上是一个返回的对象, 您可以将回调函数附加到该对象上, 而不必将回调作为参数传递给另一个函数。
 - b. .then() 链式调用 避免 回调地狱
 - c. 顺序可以得到保证
 - d. 错误处理 .catch()

e. 大多数现代Web API都是基于promise的，因此你需要了解promise才能充分利用它们。这些API包括[WebRTC](#)，[Web Audio API](#)，[Media Capture and Streams](#)等等。随着时间的推移，Promises将变得越来越重要，因此学习使用和理解它们是学习现代JavaScript的重要一步。

合作异步JavaScript: Timeouts and intervals

1. `setTimeout()`
2. `setInterval()`
3. `requestAnimationFrame()`