Promise --- 承诺

Js 同步和异步

1.单线程

1. 概念：gJavaScript是一门单线程的语言，因此，JavaScript在同一个时间只能做一件事，单线程意味着，如果在同个时间有多个任务的话，这些任务就需要进行排队，前一个任务执行完，才会执行下一个任务
2. 为什么是单线程

其实，JavaScript的单线程，与它的用途是有很大关系，我们都知道，JavaScript作为浏览器的脚本语言，主要用来实现与用户的交互，利用JavaScript，我们可以实现对DOM的各种各样的操作，如果JavaScript是多线程的话，一个线程在一个DOM节点中增加内容，另一个线程要删除这个DOM节点，那么这个DOM节点究竟是要增加内容还是删除呢？这会带来很复杂的同步问题，因此，JavaScript是单线程的

1. 同步任务和异步任务

（1）为什么会有同步和异步

  因为JavaScript的单线程，因此同个时间只能处理同个任务，所有任务都需要排队，前一个任务执行完，才能继续执行下一个任务，但是，如果前一个任务的执行时间很长，比如文件的读取操作或ajax操作，后一个任务就不得不等着，拿ajax来说，当用户向后台获取大量的数据时，不得不等到所有数据都获取完毕才能进行下一步操作，用户只能在那里干等着，严重影响用户体验

  因此，JavaScript在设计的时候，就已经考虑到这个问题，主线程可以完全不用等待文件的读取完毕或ajax的加载成功，可以先挂起处于等待中的任务，先运行排在后面的任务，等到文件的读取或ajax有了结果后，再回过头执行挂起的任务，因此，任务就可以分为同步任务和异步任务

（2）同步任务

同步任务是指在主线程上排队执行的任务，只有前一个任务执行完毕，才能继续执行下一个任务，当我们打开网站时，网站的渲染过程，比如元素的渲染，其实就是一个同步任务

（3）异步任务

异步任务是指不进入主线程，而进入任务队列的任务，只有任务队列通知主线程，某个异步任务可以执行了，该任务才会进入主线程，当我们打开网站时，像图片的加载，音乐的加载，其实就是一个异步任务

1. 异步机制

JavaScript的异步机制包括以下几个步骤

（1）所有同步任务都在主线程上执行，行成一个执行栈

（2）主线程之外，还存在一个任务队列，只要异步任务有了结果，就会在任务队列中放置一个事件

（3）一旦执行栈中的所有同步任务执行完毕，系统就会读取任务队列，看看里面还有哪些事件，那些对应的异步任务，于是结束等待状态，进入执行栈，开始执行

（4）主线程不断的重复上面的第三步

异步 ：操作之间没有什么关系，同时进行多个操作

同步 ：同时只能做一件事

异步：代码比较复杂

同步：代码比较简单