JavaScript ES6 Promises 使异步任务的处理方式变成线性， 这是大多数现代Web应用程序中的一项任务。 而不是依靠回调函数 —— 通过JavaScript框架（如jQuery）普及。JavaScript Promises 使用一个中心直观的机制来跟踪和响应异步事件。它不仅使调试异步代码变得更容易，而且使得编写它也是一种乐趣。

所有 JavaScript Promise 都是通过 Promise() 构造函数开始和结束。

在内部使用 resolve() 和 reject() 方法，当一个 Promise 被完成或拒绝时，我们可以分别向一个 Promise 对象发出信号。 then() 与 catch() 方法随后可以被调用，用以处理完成或拒绝 Promise 后的工作。

除了 JavaScript Promise，异步函数进一步重写了传统的异步代码结构，使其更具可读性。每当我向客户展示带有async 编程功能的代码时，第一个反应总是令人惊讶，随之而来的是了解它是如何工作的好奇心。

一个异步函数由两部分构成:

1) 一个以 async 为前缀的常规函数

2) 在异步函数（Async function）内，使用 await 关键字调用异步操作函数