

校企合作项目记录表（跟岗学习专用）

记录时段	2019 年 12 月 19 日 —— 2019 年 12 月 21 日	
企业名称	广东元心科技有限公司	
记录类型	学习记录	
学生姓名	苏俊贤	指导老师签名：
记录内容 (可另页附)	<p style="text-align: center;">ES6 Promise</p> <p>1. 什么是 Promise</p> <p>Promise 是一种异步编程的解决方案，其实就是一个构造函数，它身上有 all、reject、resolve 这几个方法，原型上有 then、catch 等方法。</p> <p>Promise 有以下两个特点：</p> <p>(1) 对象的状态不受外界影响。Promise 对象代表一个异步操作，有三种状态：pending（进行中）、fulfilled（已成功）和 rejected（已失败）。只有异步操作的结果，可以决定当前是哪一种状态，任何其他操作都无法改变这个状态。这也是 Promise 这个名字的由来，它的英语意思就是“承诺”，表示其他手段无法改变。</p> <p>(2) 一旦状态改变，就不会再变，任何时候都可以得到这个结果。Promise 对象的状态改变，只有两种可能：从 pending 变为 fulfilled 和从 pending 变为 rejected。只要这两种情况发生，状态就凝固了，不会再变了，会一直保持这个结果，这时就称为 resolved（已定型）。如果改变已经发生了，你再对 Promise 对象添加回调函数，也会立即得到这个结果。这与事件（Event）完全不同，事件的特点是，如果你错过了它，再去监听，是得不到结果的。</p> <p>有了 Promise 对象，就可以将异步操作以同步操作的流程表达出来，避免了层层嵌套的回调函数。此外，Promise 对象提供统一的接口，使得控制异步操作更加容易。</p> <p>Promise 构造函数接受一个函数作为参数，该函数的两个参数分别是 resolve 方法和 reject 方法。</p> <p>如果异步操作成功，则用 resolve 方法将 Promise 对象的状态，从「未完成」变为「成功」（即从 pending 变为 resolved）；</p> <p>如果异步操作失败，则用 reject 方法将 Promise 对象的状态，从「未完成」变为「失败」（即从 pending 变为 rejected）。</p>	

2. Promise 的一些方法

Promise.resolve()

resolve 就是成功的时候的回调，他把 promise 的状态修改为 resolved，这样我们在 then 中就能捕捉到，然后执行“成功”情况的回调。

Promise.reject()

reject 就是失败的时候的回调，他把 promise 的状态修改为 rejected，这样我们在 then 中就能捕捉到，然后执行“失败”情况的回调。

Promise.prototype.then()

Promise 实例具有 then 方法，也就是说，then 方法是定义在原型对象 Promise.prototype 上的。它的作用是为 Promise 实例添加状态改变时的回调函数。前面说过，then 方法的第一个参数是 resolved 状态的回调函数，第二个参数（可选）是 rejected 状态的回调函数。

Promise.prototype.catch()

Promise.prototype.catch 方法是 .then(null, rejection) 或 .then(undefined, rejection) 的别名，用于指定发生错误时的回调函数。

Promise.all()

与 then 同级的另一个方法，all 方法，该方法提供了并行执行异步操作的能力，并且在所有异步操作执行完后并且执行结果都是成功的时候才执行回调。