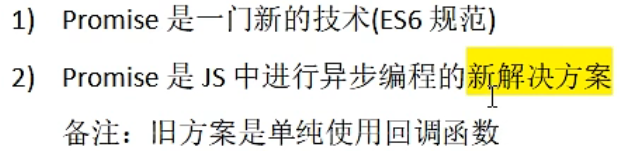
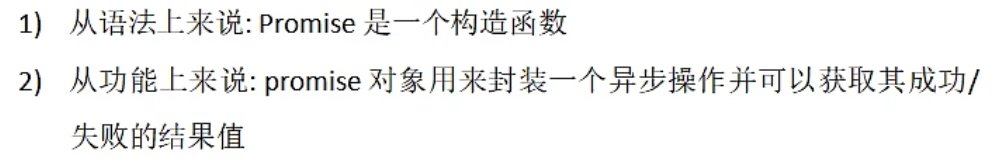
### Promise概念

#### 什么是promise

##### 抽象表达



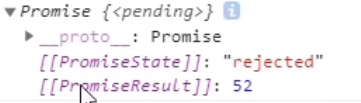
##### 具体表达



#### 为什么要用promise

* 指定回调函数的方式更加灵活。
* 支持链式调用，可以解决回调地狱问题。

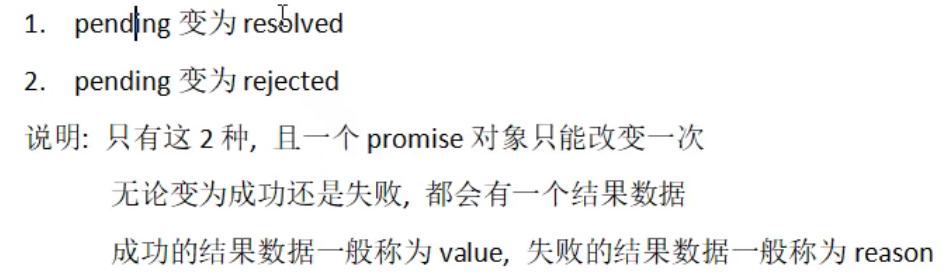
#### Promise对象属性



##### Promise的状态改变

promise的状态是实例对象的一个属性PromiseState。有三个值：pending\resolved(fullfilled)\reject。

状态变化只有两种情况：

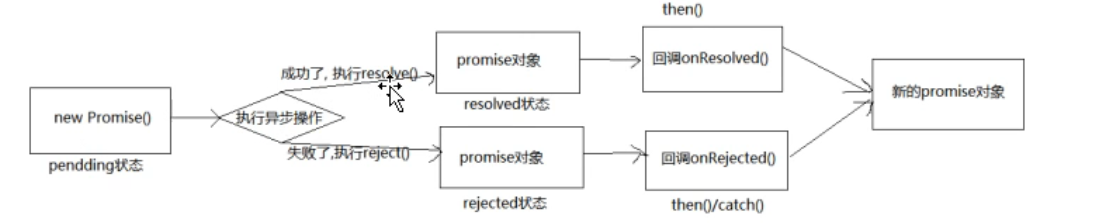


##### PromiseResult

保存的是异步任务成功或失败的结果。

只有resolve和reject函数能修改该属性的值。

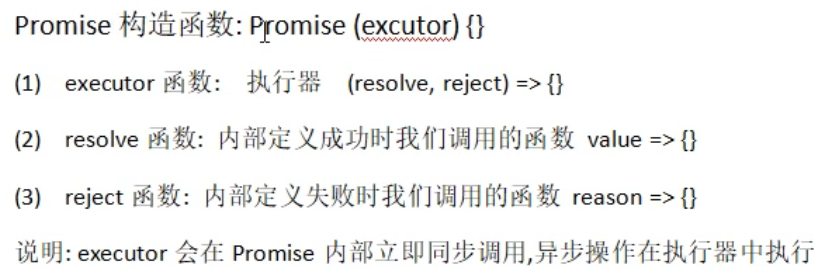
#### Promise的工作流程



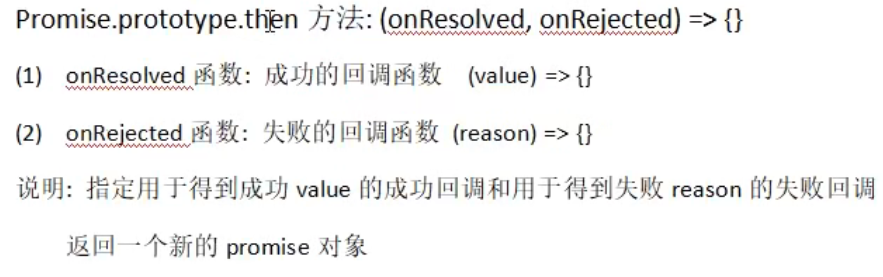
也就是说当new Promise(excutor(){})时，Promise的状态为pendding状态，此时excutor(){}是同步执行的，然后直接产生resolved/rejected状态，调用then方法时，then方法的两个形参都是监听器，根据Promise的状态执行对应的形参函数，最后返回一个新的Promise对象。

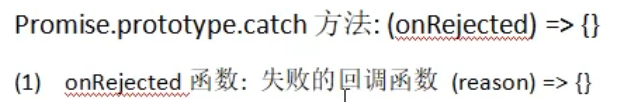
#### 如何使用promise

##### API

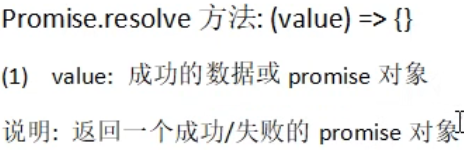


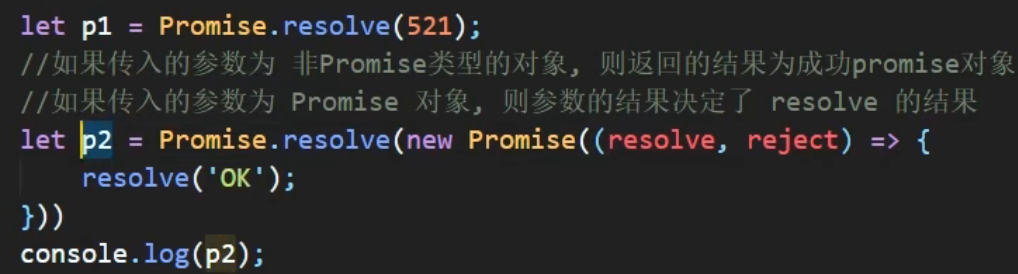
以下是属于promise实例对象的API：

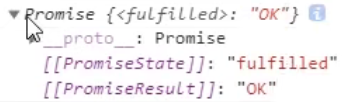


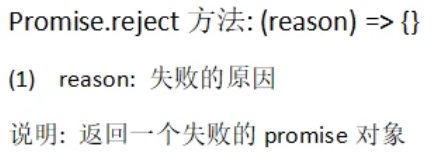


以下是属于promise构造函数对象的API：

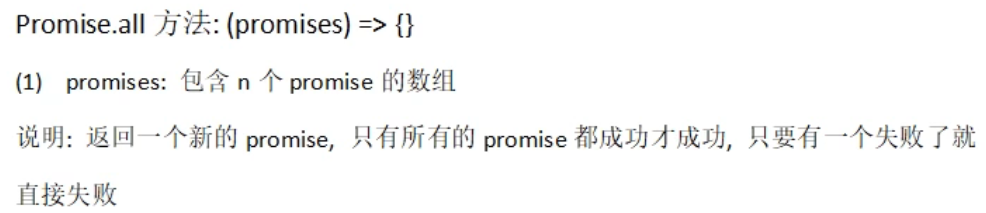


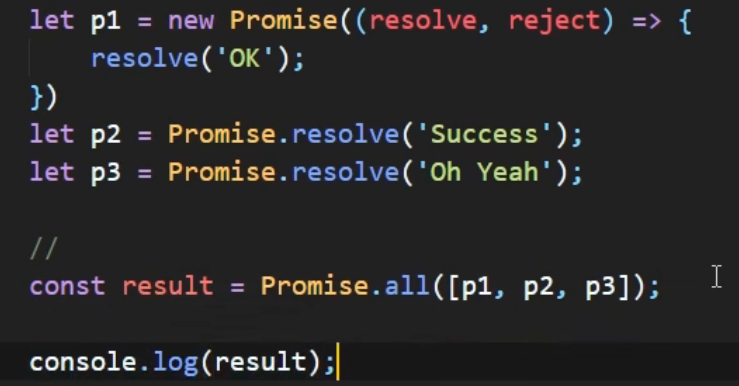


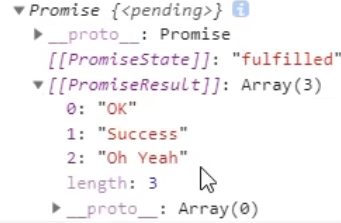




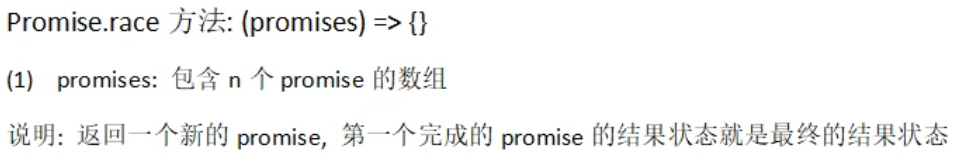
不管传入什么参数，返回的PromiseState都是rejected，PromiseResult是传入的参数。





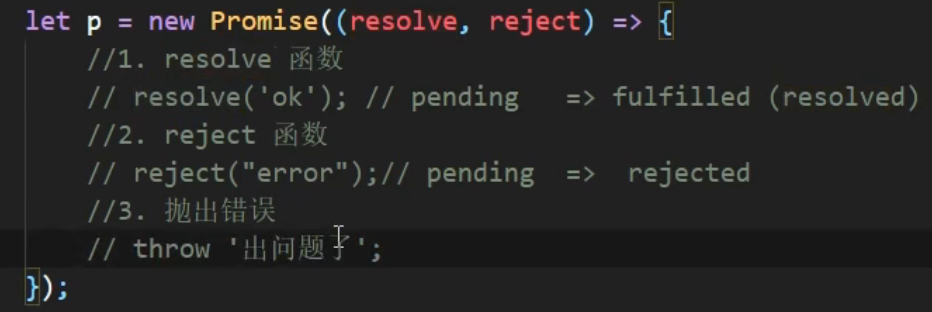


如果promises中有失败的promise，PromiseState为rejected，PromiseResult为失败的promise的结果。

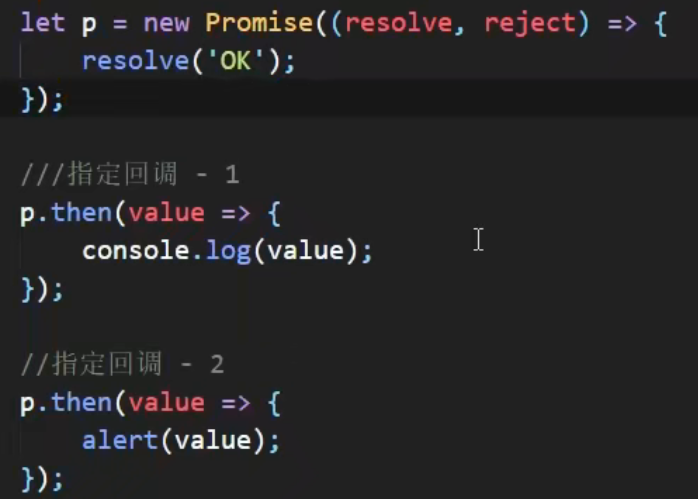


#### Promise的几个关键问题

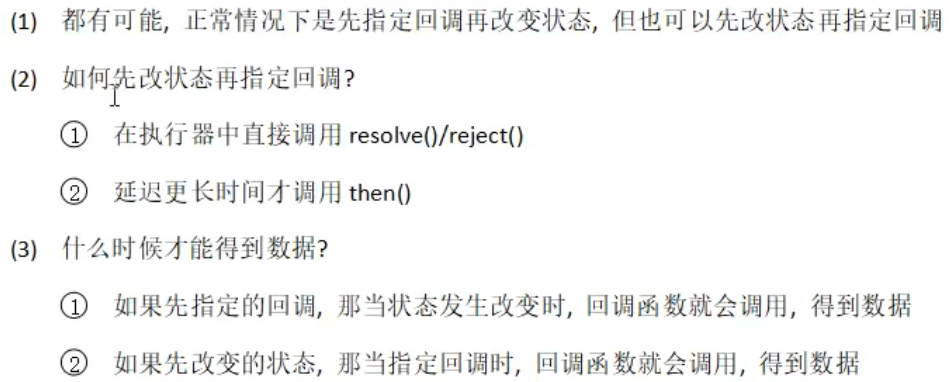
##### 如何改变promise对象状态（3种方式）



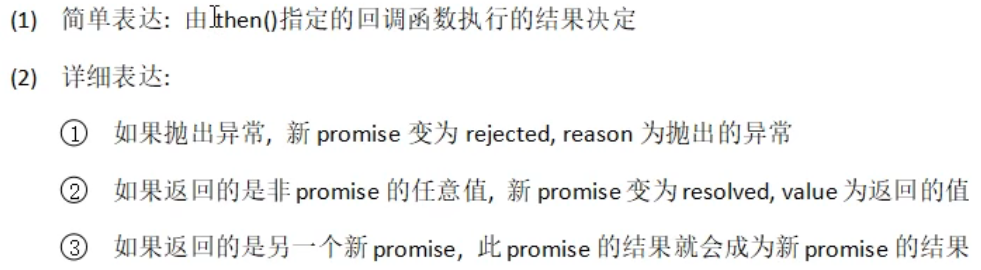
##### Then方法能否多次指定回调（能）



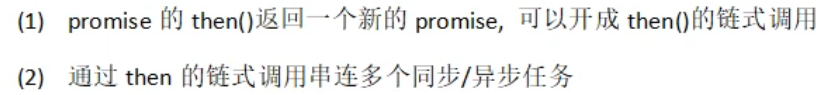
##### 改变状态和回调（例如then方法）谁先执行



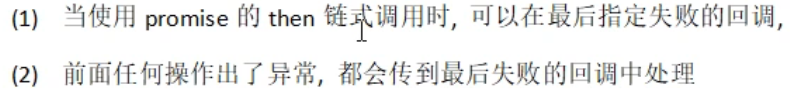
##### Then方法返回的promise有什么样的特点

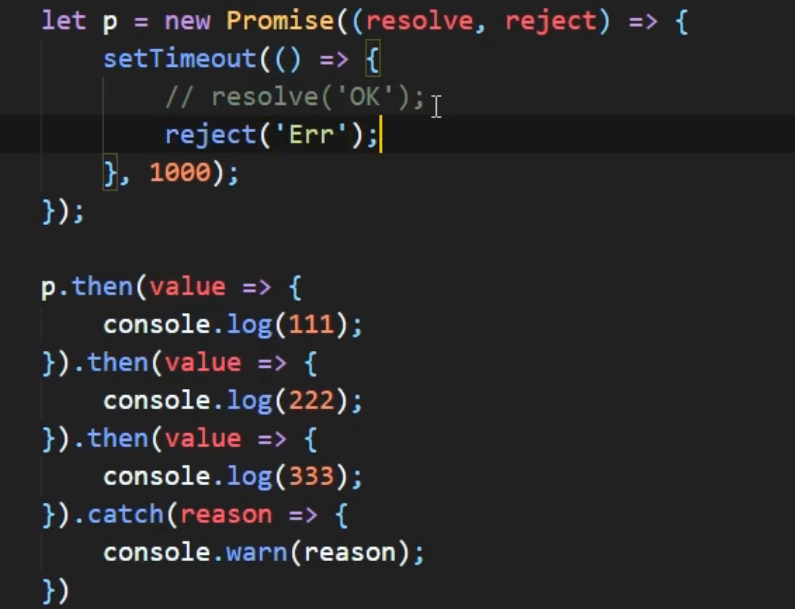


##### Promise如何串联多个操作任务（promise链）



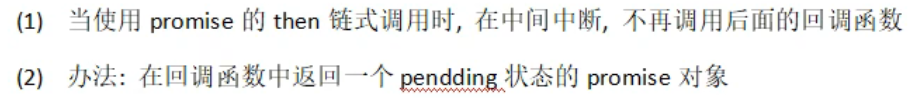
##### Promise异常穿透

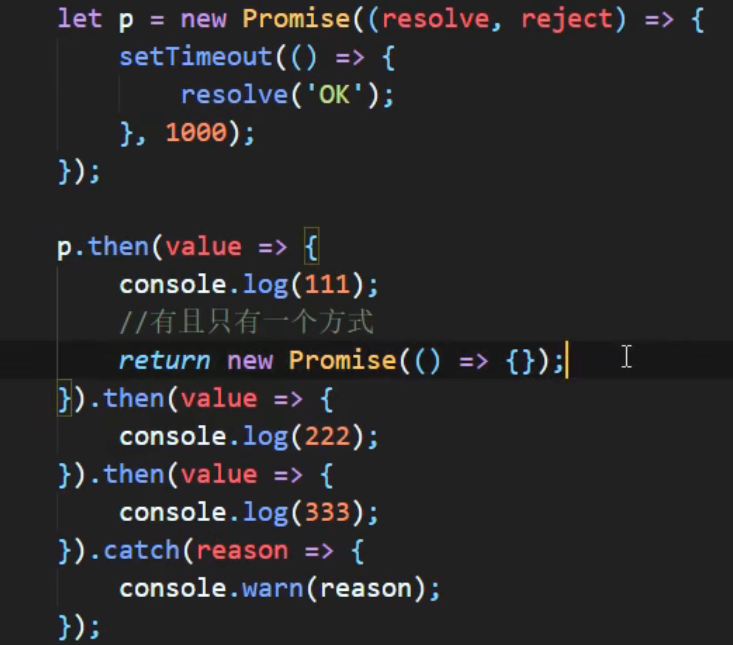






##### 中断promise链





### Promise的使用

#### Promise初体验

这是一个随机抽取奖品的promise构造函数。

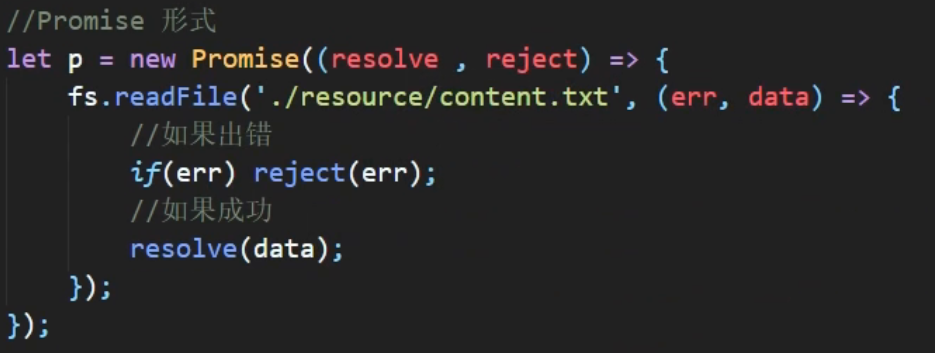


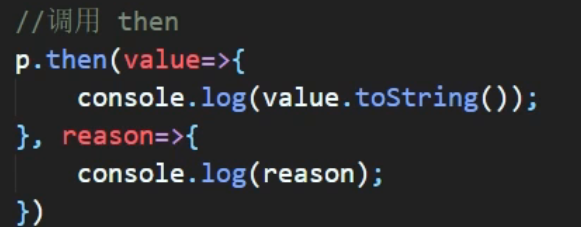


resolve，reject两个函数代表成功/失败时将返回的结果值。

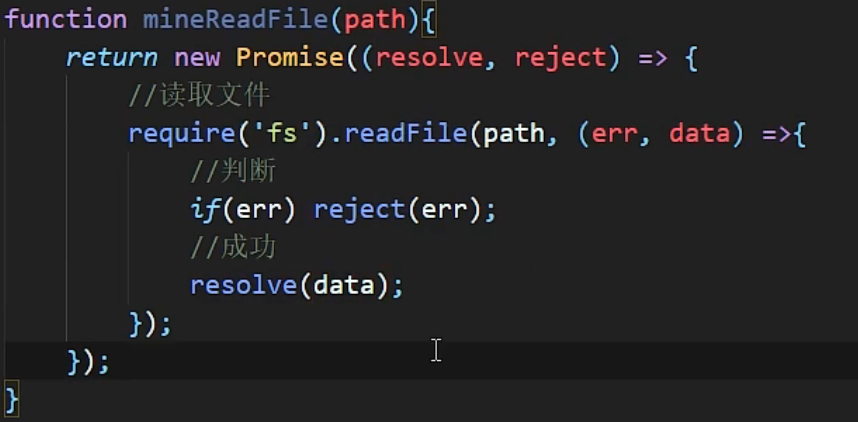
#### Promise实践练习

##### Fs读取文件操作封装



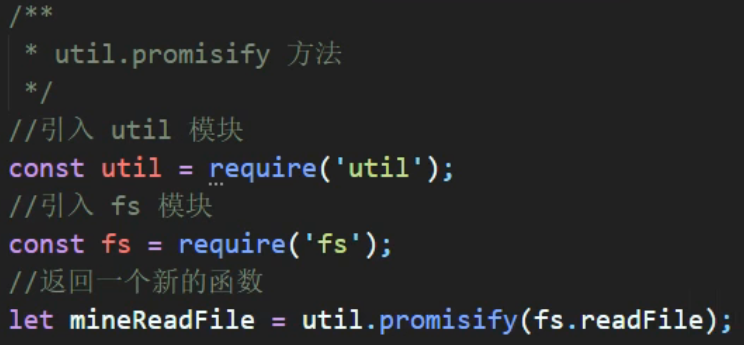


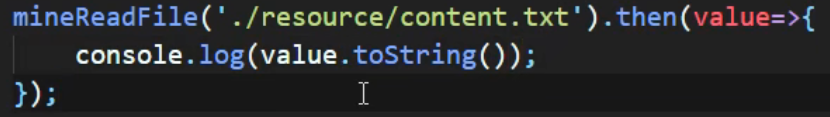
###### 将FS操作封装成一个函数，返回值为Promise对象



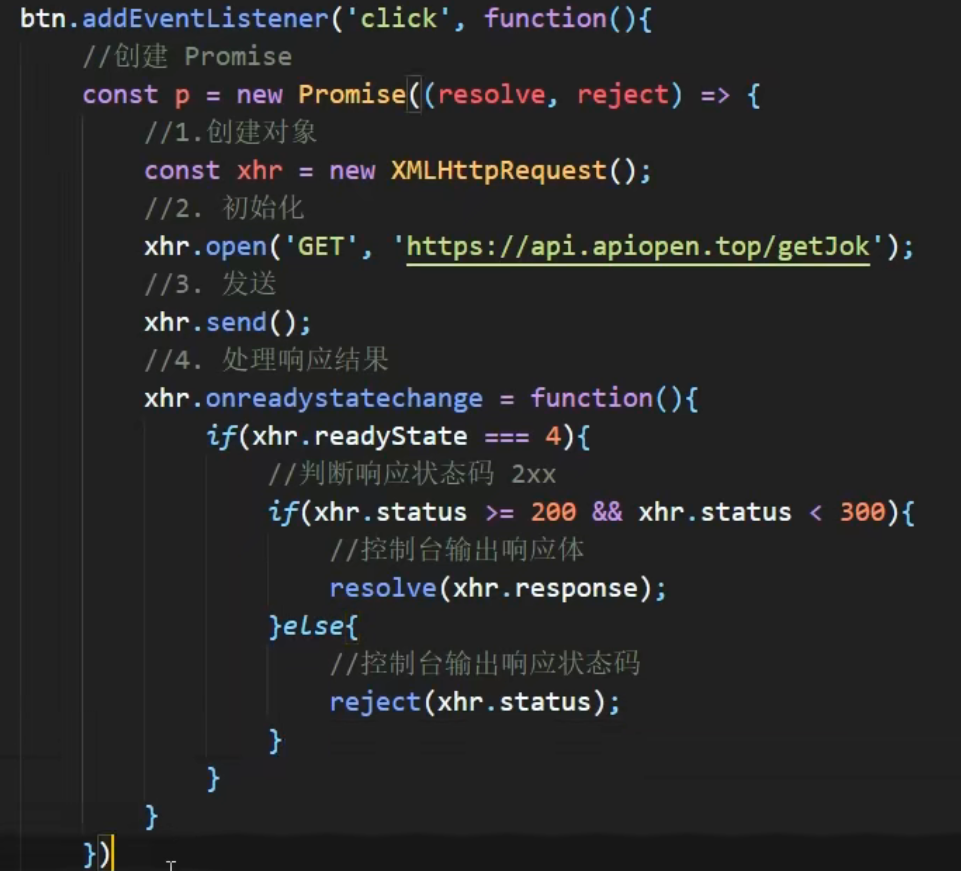


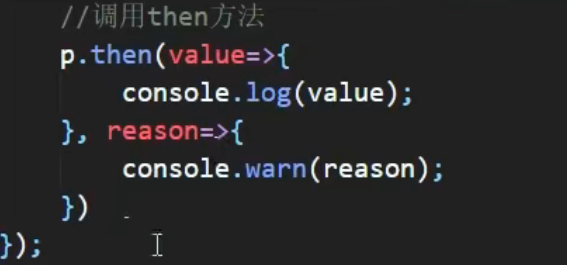
使用util模块一键封装能返回promise对象的fs函数



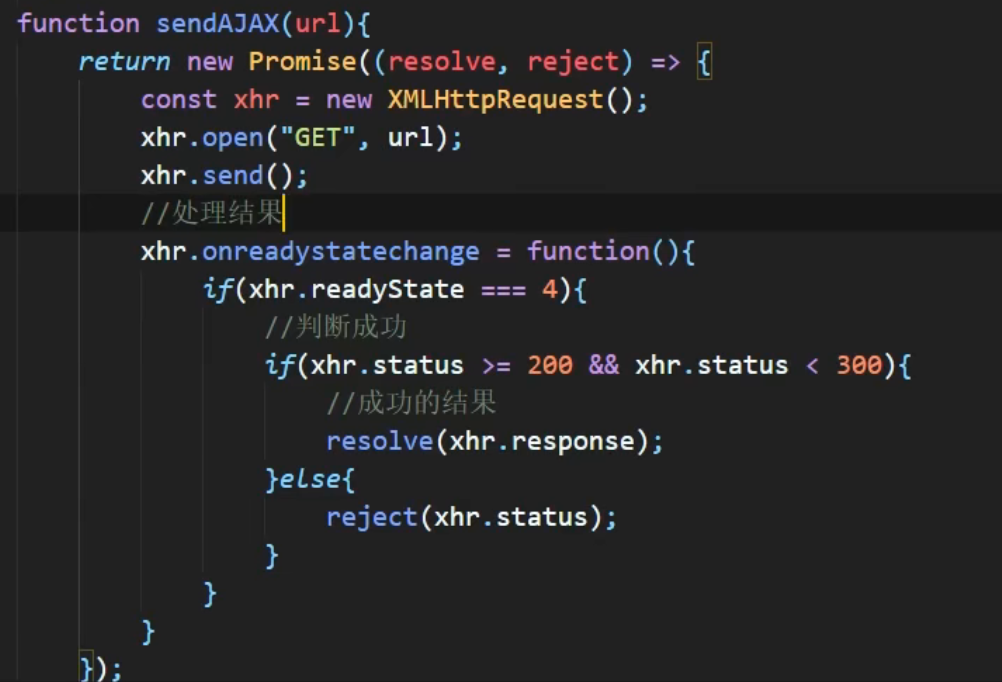


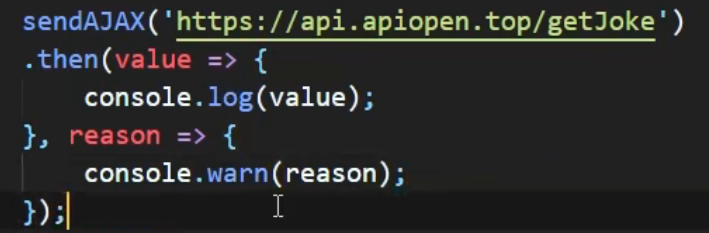
##### Ajax操作封装





###### 将AJAX操作封装成一个函数，返回值为Promise对象(axios原理)





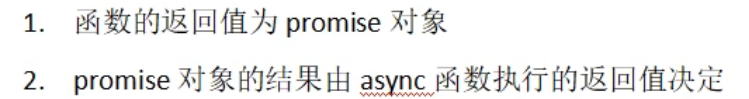
### 自定义（手写）promise

结合promise的几个关键问题去写。

* 初始结构搭建
* 实例方法resolve与reject的实现
* throw抛出异常改变状态
* promise状态只能修改一次
* then方法执行回调（模拟事件监听（添加callback保存回调））
* 指定多个回调（then多次赋值）
* 同步/异步then方法返回结果实现
* 封装catch方法与异常穿透
* 封装resolve/reject静态方法
* 封装all方法
* 封装race方法
* then方法异步执行

### async与await

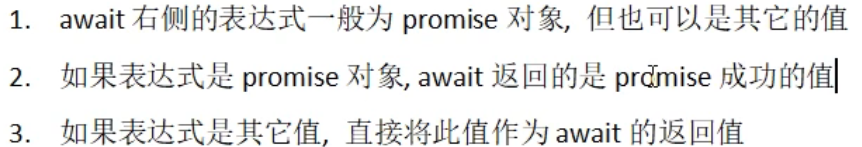
#### async函数



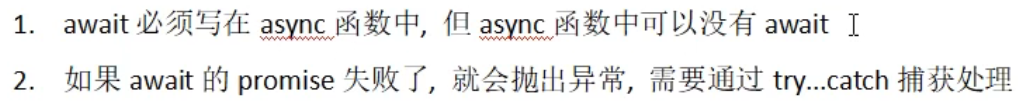
Async方法返回的promise结果规则与then方法返回的类似。

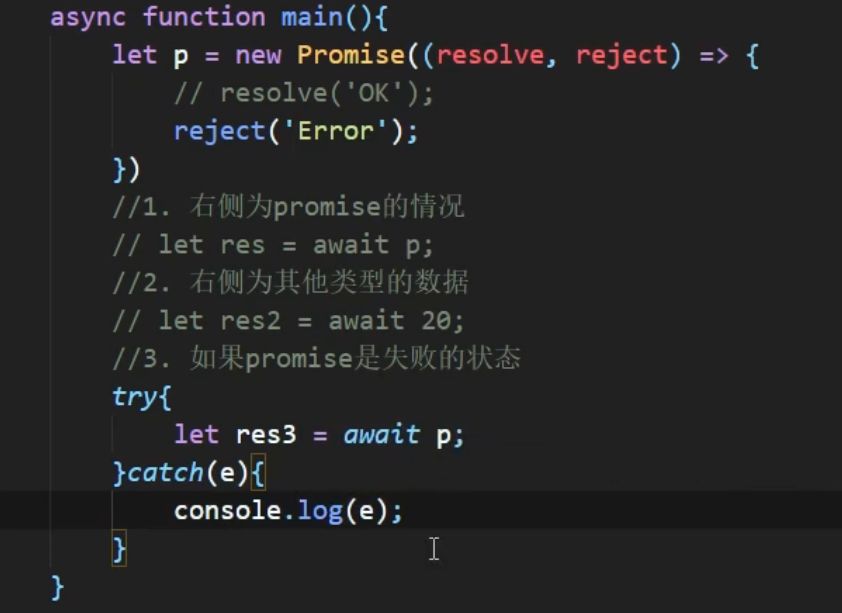


#### Await表达式



##### 注意

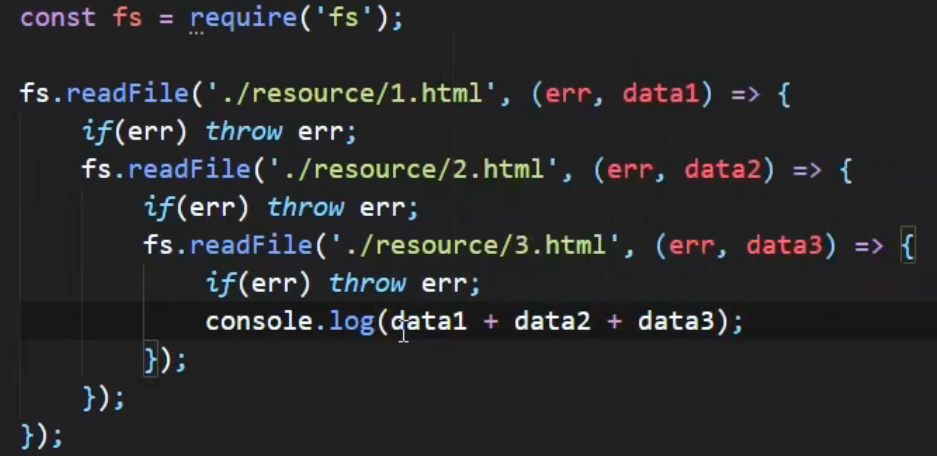




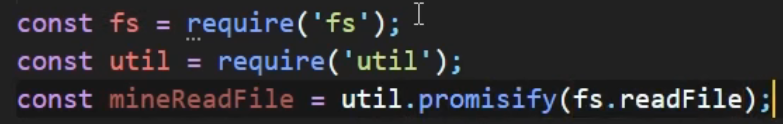
#### async与await结合

##### 实现读取三个html文件

###### 原始方法（回调函数）



###### async与await的方法





##### 发送AJAX请求

