学习记录

1. 继续学习promise 及相应使用场景

then方法可以返回promise对象 所以意味着可以进行链式操作 而后面都需要前面执行完毕才会执行这也是我们开发中会遇到的情况

而promise.prototype.catch 等于 .then(null , function(){}) 是在rejected的情况下触发的函数
而promise.catch方法可以进行错误的捕获即使没有reject()调用 依然会执行 但是如果是then(function(){} , function(){})这样写就不会有捕获错误的效果

而这个效果可捕获之前的所有的错误同步异步均可 却无法捕获之后所定义的then等出现的错误

promise.finally() 方法 是无论resolve 还是reject 都会最后执行的函数 但是无法获取对应之前的状态

promise.all([...]).then().catch() 方法是参数promise对象全部fulfilled成功状态时才会触发then方法 或者有一个变为rejected是触发catch方法

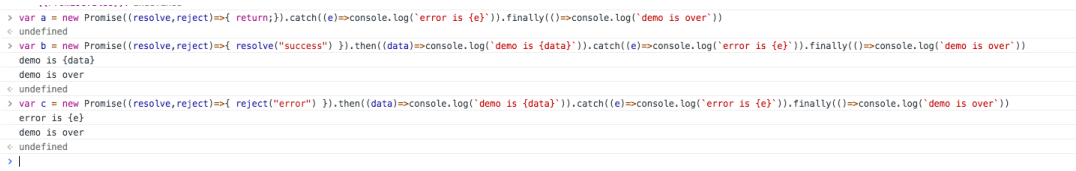
- 如果内部promise对象包含catch方法一旦被rejected并不会触发promise.all方法的catch方法

已解决
1. return ; Promise将不会执行then catch finally方法

```
var a = new Promise((resolve,reject)=>{
return;}).catch((e)=>console.log(`error is {e}`)).finally(()=>console.log(`demo is over`))

var b = new Promise((resolve,reject)=>{ resolve("success")
}).then((data)=>console.log(`demo is {data}`)).catch((e)=>console.log(`error is {e}`)).finally(()=>console.log(`demo is over`))
```

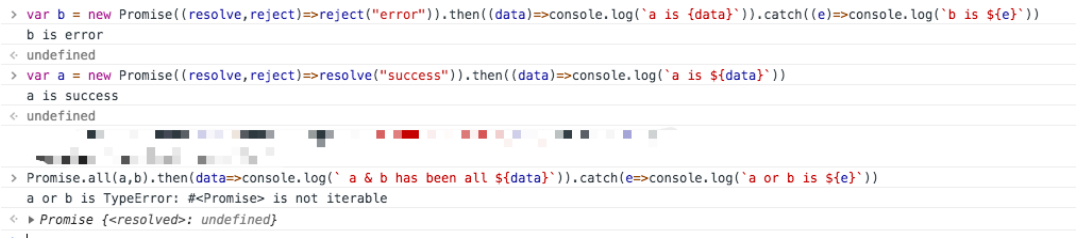
var c = new Promise((resolve,reject)=>{ reject("error") }).then((data)=>console.log(`demo is {data}`)).catch((e)=>console.log(`error is {e}`)).finally(()=>console.log(`demo is over`))



2. then方法虽然是返回promise对象的 但是触发必须需要调用resolve() 而catch除了在reject（）触发也可以在报错error时触发

3. promise.all 方法如果只是全fulfileed状态then 有一个rejected就catch的话 岂不是异步操作会一直catch？ 试验一下

- 如果b rejected 那么Promise.all 方法 catch不会触发成功



未解决