1. 在ES5中，如何实现多个形参不确定，还要求参数值与形参变量一一对应?

答: function order(menu){

//menu: { zs:xxx, tang:xxx, td:xxx, rc:xxx, sc:xxx}

menu.zs...

menu.tang...

}

order({ zs:xxx, tang:xxx, td:xxx, rc:xxx, sc:xxx})

2. function Fun(){

var i=0;

return function(){

console.log(i++);

}

}

var f1=Fun();

var f2=Fun();  
 f1(); //0

f1(); //1

f2(); //0 不是2?

f2(); //1 不是 3?

答: 妈妈分两次分别生了两个孩子，每次生孩子都给孩子包一个专属的红包。所以，f2和f1用的不是一个红包。他们分别有自己专属的红包。

3. .then()中的函数为什么不加()也能执行？谁调用的？

答：liang().then(ran)，其实还是回调函数的方式，先把ran放入.then中留着备用。只有当前一个liang中调用了door()打开门后，才自动找到.then()中提前放好的ran，此时才自动加()调用。

4. liang().then(ran).then(dong).then(liang)为什么不会形成死循环?

答: 每次调用一个函数，都是创建并返回一个新的格子间。新的格子间，如果后边没有接then,就不会有下一步后续任务。

这里liang() ... .then(liang)等于先后调用了两次liang，返回过两个格子间。而第二次调用liang返回的格子间绝对不等于第一个格子间，且第二个格子间后边没有.then。所以，第二次调用.then(liang)后，就没有后续步骤了。

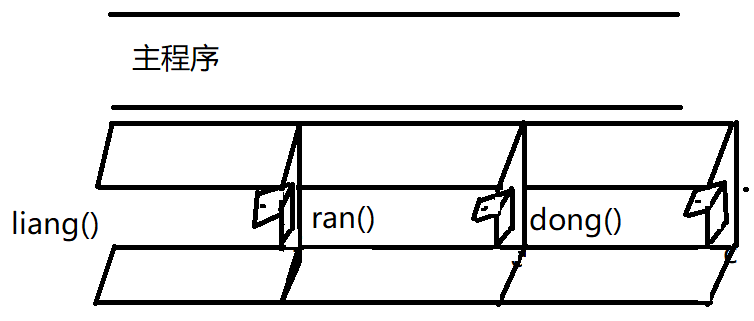
5. ran不加()也能调用，是.then()的功劳吗？

答？不只是then。

所有.then,其实起到的是收集函数并排队的作用。比如: liang().then(ran).then(dong)，等效于: [liang, ran, dong]顺序存储好。然后，依次自动调用每一个函数，如果前一个函数调用了door，才i++，自动调用第二个。如果第二个调用了door，才i++，自动调动第三个。所以ran的执行，是因为被.then()收齐起来，再由上一步调用door()共同触发执行的。

6. console.log(“比赛结束”)，不放在then中，依然会在所有人到达终点前就执行？

答：是的！看图



Promise格子间本身都是异步的。第一个格子间是异步的，在主程序之外执行的，所以，就算后续格子间与第一个格子间是串联的关系，无非也是在主程序之外，串联起一个更长的异步执行过程而已。所以，由promise串起来的多个异步函数，组成的是一个大的异步任务，但与主程序相比，依然是异步的。

7. var bang=await liang(); 可不可以不加await

答: 不加虽然不会报错，但是后续程序不会等待当前函数执行完，就提前执行了。执行顺序是不是颠倒了！

8. return new Promise(

function(resolve, reject){

}

)中，resolve和reject，必须叫这两个名字吗？

答: 只要是形参，永远没有固定的名称

但是，顺序是固定的！

将来只要调用第1个开关，都表示当前函数执行成功，都走.then()

将来只要调用第2个开关，都表示当前函数执行错误，进入.catch()

9. Uncaught (in promise) 说明是人家Promise函数中用reject()抛出了错误！但是你没接住:

答: ES6: .catch(e=>console.log(e))

ES7: try{

...

}catch(e){ console.log(e) }

10. 异步是不是脱离开~~全局~~主程序，没有和主程序在一起? 如果加上await，是不是就等待了呢？

答: 问题一: 异步的确是不和主程序一起执行的

问题二: await不能随便用:

1. 这个异步函数，必须支持Promise  
 2. await外围必须套上(async function(){...})();

11. 到目前为止有哪些函数是异步的:

答: setTimeout() setInterval() 可以自己控制结束的时间

xhr.onreadystatechange=function(responseText){

}

将来:

ajax({url, type, data, dataType}).then(res=>{...})

jquery:

$.ajax({url, type, data, dataType}).then(res=>{...})

vue:

this.axios.get(url,{ params: ... }).then(res=>{ ... })

nodejs:

pool.query(sql,[...],(err,result)=>{//异步

... ...

pool.query(sql,[...],(err,result)=>{//异步

... ...

pool.query(sql,[...],(err,result)=>{//异步

... ...

res.send(result)

})

})

})

11. 如果等待多个异步任务都完成后，才执行后续操作，用ES7怎么写？如果Promise.all中抛出异常，如何处理?

答: var arr=await Promise.all([liang(), ran(), dong() ])

console.log("比赛结束")

其中: arr中: [101,102,103]

如果Promise.all中抛出异常，则依然使用try{ ... }catch(e){ ... }处理