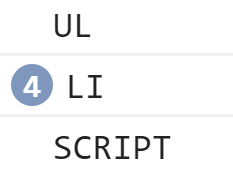
1. Promise.all怎么能即使某个任务出错，也能接受其他执行正确的任务的返回值呢？

答：不能。因为all中的任务和then是前置条件和后置步骤的关系。前置条件，只要有一个不满足，表示整件事儿，办不了。比如，去办手续，人家说要提供身份证，户口本，照片。那么，这三样，缺一样，这个手续的手续步骤都是办不成的。所以，前置条件只是部分成立，是没有意义的。

从程序角度说，.then和.catch()是二选一的关系。只要有人开了err()，状态变为reject，整个任务就会进入.catch()，.then就不再执行。不可能又执行.then()，又执行.catch()。

2. 为什么第一次输出

刷新后变成: 

答: 控制台有个习惯，相邻的多个相同的输出结果，只显示一个，在左侧圆圈中标出重复的次数。

但是，本例中，我们不应该关心控制台的显示格式。而应该只关心总次数对不对。比如: 根据深度优先遍历，ul到script之间，总共应该输出4个li。所以，以上两种方式都是正确的！

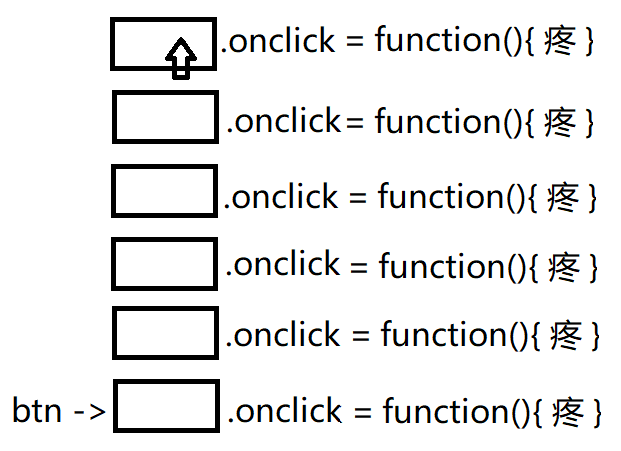
3. for(var btn of btns){  
 btn.onclick=function(){

alert("疼")

}  
 }

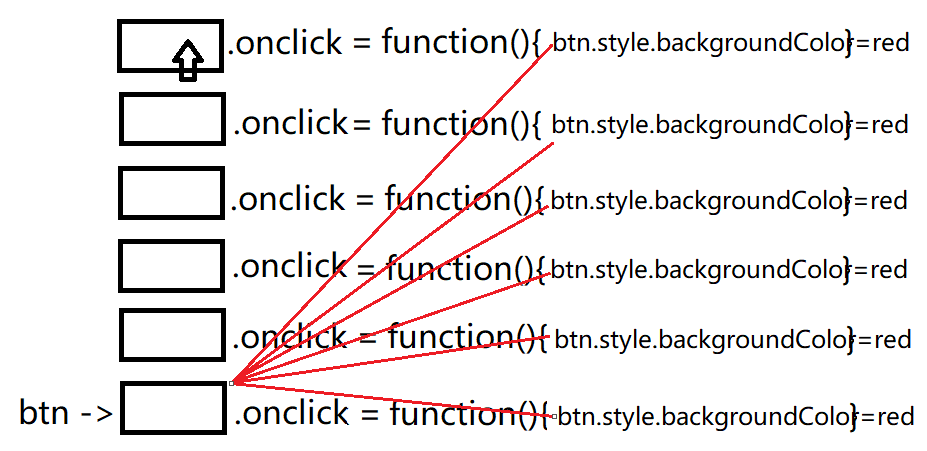
是不是也只能最后一个按钮喊疼？

答: 不是。每个按钮都能喊疼！

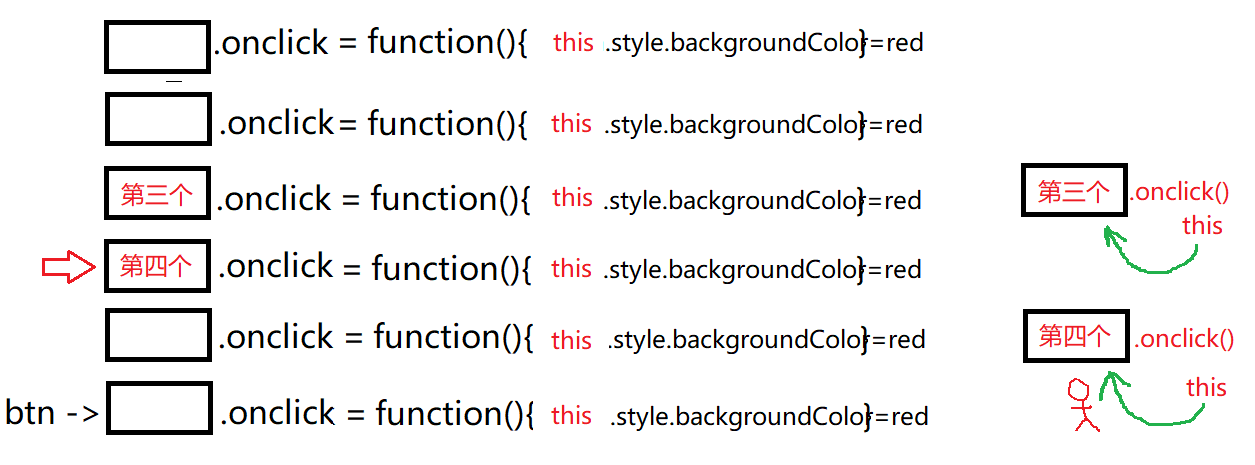


为什么onclick中改成 btn.style.backgroundColor=red就只有最后一个按钮才能变红呢？

因为处理函数中没有局部变量，所有函数中的btn都指向全局变量btn，也就是最后一个按钮：



为什么用this又对了呢？



4. var n=parseInt(span.innerHTML)为什么不用Number()转换

答: parseInt()和parseFloat()是专门将字符串转为数字类型的函数。且parseInt()和parseFloat()可自动去掉数字后非数字字符，比如px单位。而Number()无此功能，比较弱。所以，今后，能用parseInt()或parseFloat()都用parseInt()和parseFloat()。

5. if(btn.innerHTML=="+"){

n++

}else if(n>1){

n--

}  
 为什么else if中不写btn.innerHTML=="-"

答: 因为if else结构中，如果if条件不满足，则已经隐含了btn.innerHTML不是+，在这个例子中，是有两种情况: btn是+或btn是-，那么只要不是+，就意味着一定是-。所以，else if中不用重复写btn.innerHTML=="-"。

6. 到目前为止所有this的用法:

答: this的汇总:

1. 普通函数或匿名函数自调中的this->window

严格模式下: this->undefined

2. obj.fun() fun中的this->obj

3. new Fun() Fun中的this->正在创建的新对象

4. 原型对象中的函数里的this->将来调用当前函数的.前的当前类型的子对象。

5. btn.onclick=function(){... } this->btn

6. 回调函数:

arr.forEach(function(){ ... })

arr.map(function(){ ... })

setInterval(function(){ ... })

setTimeout(function(){ ... })

$.ajax({

...

success:function(){ ... }

}).then(function(){

...

})

this->window 所有回调函数，真正被调用时，前边是没有任何"对象."前缀

所以，通常如果希望回调函数中的this不指window，而是跟外部的this保持一致，都要改为箭头函数