Promise 是什么？

作者：方应杭
链接：https://zhuanlan.zhihu.com/p/22782675
来源：知乎

window.Promise 已经是 JS 的一个内置对象了。

1. Promise 有规格文档吗？
2. 你一般如何使用 Promise。

目前的 Promise 都遵循 Promises/A+ 规范。

英文规范： promisesaplus.com/

中文翻译：图灵社区：阅读：【翻译】Promises/A+规范

看完规范你可以了解 Promise 的全貌，本文主要讲讲 Promise 的用途。

Promise 之前的时代——回调时代

假设我们用 getUser 来说去用户数据，它接收两个回调 sucessCallback 和 errorCallback：

```
function getUser(successCallback, errorCallback){
  $.ajax({
    url:'/user',
    success: function(response){
      successCallback(response)
    },
    error: function(xhr){
      errorCallback(xhr)
    }
  })
}
```

看起来还不算复杂。

如果我们获取用户数据之后还要获取分组数组、分组详情等，代码就会是这样：

```
getUser(function(response){
    getGroup(response.id, function(group){
      getDetails(groupd.id, function(details){
        console.log(details)
      },function(){
```

```
        alert('获取分组详情失败')
      })
    }, function(){
      alert('获取分组失败')
    })
  }, function(){
  alert('获取用户信息失败')
})
```

三层回调，如果再多一点嵌套，就是「回调地狱」了。

Promise 来了

Promise 的思路呢，就是 getUser 返回一个对象，你往这个对象上挂回调：

```
var promise = getUser()
promise.then(successCallback, errorCallback)
```

当用户信息加载完毕，successCallback 和 errorCallback 之一就会被执行。

把上面两句话合并成一句就是这样的：

```
getUser().then(successCallback, errorCallback)
```

如果你想在用户信息获取结束后做更多事，可以继续 .then：

```
getUser().then(success1).then(success2).then(success3)
```

请求成功后，会依次执行 success1、success2 和 success3。

如果要获取分组信息：

```
getUser().then(function(response){
  getGroup(response.id).then(function(group){
    getDetails(group.id).then(function(){

    },error3)
  },error2)
}, error1)
```

这种 Promise 写法跟前面的回调看起来其实变化不大。
真的，Promise 并不能消灭回调地狱，但是它可以使回调变得可控。你对比下面两个写法就知道了。

```
getGroup(response.id, success2, error2)

getGroup(response.id).then(success2, error2)
```

用 Promise 之前，你不能确定 success2 是第几个参数；

用 Promise 之后，所有的回调都是

```
.then(success, error)
```

这样的形式。

以上是 Promise 的简介，想完整了解 Promise，请参考下面的自学链接。

Promise对象 -- JavaScript 标准参考教程（alpha）