

1. 如何改变promise的状态？

- (1) `resolve(value)`: 如果当前是`pending`就会变为`resolved`
- (2) `reject(reason)`: 如果当前是`pending`就会变为`rejected`
- (3) 抛出异常: 如果当前是`pending`就会变为`rejected`

2. 一个promise指定多个成功/失败回调函数, 都会调用吗？

当promise改变为对应状态时都会调用

3. 改变promise状态和指定回调函数谁先谁后？

- (1) 都有可能, 正常情况下是先指定回调再改变状态, 但也可以先改状态再指定回调
- (2) 如何先改状态再指定回调?
 - ① 在执行器中直接调用`resolve()/reject()`
 - ② 延迟更长时间才调用`then()`
- (3) 什么时候才能得到数据?
 - ① 如果先指定的回调, 那当状态发生改变时, 回调函数就会调用, 得到数据
 - ② 如果先改变的状态, 那当指定回调时, 回调函数就会调用, 得到数据

4. promise.then()返回的新promise的结果状态由什么决定？

- (1) 简单表达: 由`then()`指定的回调函数执行的结果决定
- (2) 详细表达:
 - ① 如果抛出异常, 新promise变为`rejected`, `reason`为抛出的异常
 - ② 如果返回的是非promise的任意值, 新promise变为`resolved`, `value`为返回的值
 - ③ 如果返回的是另一个新promise, 此promise的结果就会成为新promise的结果

5. promise如何串连多个操作任务？

- (1) promise的`then()`返回一个新的promise, 可以开成`then()`的链式调用
- (2) 通过`then`的链式调用串连多个同步/异步任务

6. promise错误穿透？

- (1) 当使用promise的`then`链式调用时, 可以在最后指定失败的回调,
- (2) 前面任何操作出了错误, 都会传到最后失败的回调中处理

7. 中断promise链？

- (1) 当使用promise的`then`链式调用时, 在中间中断, 不再调用后面的回调函数
- (2) 办法: 在回调函数中返回一个`pending`状态的promise对象