1. util.promisify方法进行promise风格转换

// 引入 util 模块

const util = require('util');

// 引入 fs 模块

const fs = require('fs');

// 返回一个新的函数

let mineReadFile = util.promisify(fs.readFile);

mineReadFile('./4resource/content.txt').then(value => {

    console.log(value.toString());

})

1. 状态和对象的值

**## Promise 的状态**

实例对象中的一个属性 『PromiseState』

\* pending  未决定的

\* resolved / fullfilled  成功

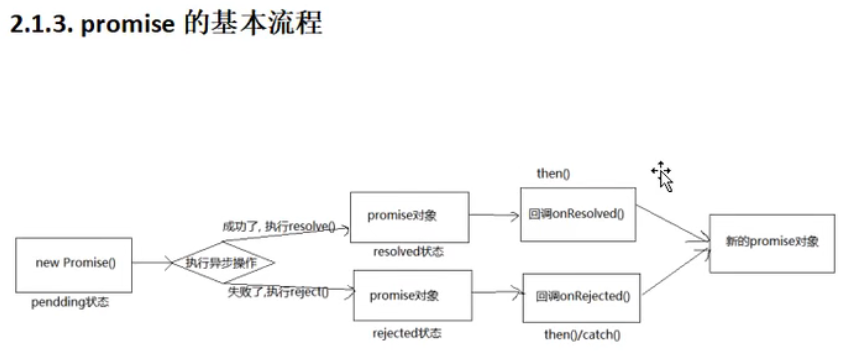
\* rejected 失败

**## Promise 对象的值**

实例对象中的另一个属性 『PromiseResult』

保存着异步任务『成功/失败』的结果

1. 执行过程



1. 同步调用

        let p = new Promise((resolve, reject) => {

            // \*\* 同步调用的

            reject('error');

        })

1. 中断

p.then(value => {

            return new Promise(() => {})

        })

1. 自定义Promise