

Promise 规范及应用：



一. 前端之殇 – 同步 vs 异步

1. 为何会出现同步&异步的问题？

a. 视图 vs 逻辑

b. 请求 + 渲染

2. 浏览器处理机制，如何解决异步问题 ——

callback

Callback 的对与错

3. 面试题中的异步

Promise – 在回调漩涡中挣扎出来的救生圈

所需知识储备：

1.异步概念

2.Callback 写法

3.Promise 基础使用

参考文章：

<http://nodejs.cn/learn/javascript-asynchronous-programming-and-callbacks>

二. Promise 简介：

1. Promise 是什么

a. 什么是 Promise ？

更加和谐的回调、更简单的处理

b. 作用与场景 – 使用 promise 场景

c. Promise 状态与状态的流转 – 类状态机

多状态流转（考察重点）

2. Promise/A+ 规范简介

a. 包含概念与术语

参数 + 状态

b. 规则

存储规则、流转规则

c. 常见规范面试题

3.Promise 常用方法

a. Promise.all

b. Promise.race

4.手写 Promise

- c. 基础框架 + 基础功能实现
- d. 丰富功能体系
- e. 边缘检测与面试中常见追问
- f. 通常面试中手写题的关注点

所需知识储备：

1. 了解 promise 使用方法
2. Promise/A+ 协议
3. Promise.all Promise.race 功能

参考文章：

<https://juejin.cn/post/6844903767654023182>

三、ESNext 时代的新解法

1. Generator 的步进

a. 步进的实现

`next()`

b. 自动化步进的解决方案

循环处理

终止标记

2. Async / Await 拯救割裂的逻辑

a. 使用方式

无需回调的异步

搭配使用

b. 本质 – Promise 的另一种存在形式

所需知识储备：

- 1.了解 generator 使用
- 2.了解步进的使用场景
- 3.了解 async、await 使用方式

参考文章：

- 1.https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Generator
- 2 . https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Statements/async_function