99 请手写实现一个 promise

> https://segmentfault.com/a/1190000013396601

100 说说从输入URL到看到页面发生的全过程，越详细越好

- 首先浏览器主进程接管， 开了一个下载线程。
- 然后进行HTTP请求（DNS查询 、IP寻址等等)， 中间会有三次捂手， 等待响应， 开始下载响应报文。
- 将下载完的内容转交给Renderer进程管理。
- Renderer进程开始解析css rule tree和dom tree， 这两个过程是并行的，所以一般我会把link标签放在页面顶部。
- 解析绘制过程中， 当浏览器遇到link标签或者script、img等标签， 浏览器会去下载这些内容， 遇到时候缓存的使用缓存，不适用缓存的重新下载资源。
- css rule tree和dom tree生成完了之后， 开始合成render tree， 这个时候浏览器会进行layout， 开始计算每一个节点的位置，然后进行绘制。
- 绘制结束后， 关闭TCP连接， 过程有四次挥手

101 描述一下 this

> this ， 函数执行的上下文， 可以通过 apply ， call ， bind 改变 this 的指向 。对于匿名函数或者直接调用的函数来说，this指向全局上下文（浏览器为window， NodeJS为 global ）， 剩下的函数调用，那就是谁调用它， this 就指向谁 。当然还有es6的箭头函数， 箭头函数的指向取决于该箭头函数声明的位置，在哪里声明， this 就指向哪里

102 说一下浏览器的缓存机制

> 浏览器缓存机制有两种，一 种为强缓存，一 种为协商缓存

- 对于强缓存， 浏览器在第一次请求的时候，会直接下载资源，然后缓存在本地， 第二次请求的时候， 直接使用缓存。