19 | Promise: 使用Promise, 告别 回调函数

20 | async-await:使用同步的方式 去写异步代码

21 | Chrome开发者工具:利用网络 面板做性能分析

22 | DOM树: JavaScript是如何影响DOM树构建的?

23 | 渲染流水线: CSS如何影响首 次加载时的白屏时间?

24 | 分层和合成机制:为什么CSS 动画比JavaScript高效?

25 | 页面性能:如何系统地优化页面?

26 | 虚拟DOM: 虚拟DOM和实际的DOM有何不同?

27 | 渐进式网页应用 (PWA) : 它 究竟解决了Web应用的哪些问题?

28 | WebComponent:像搭积木一样构建Web应用

29 | HTTP-1: HTTP性能优化

30 | HTTP-2: 如何提升网络速度?

31 | HTTP-3: 甩掉TCP、TLS的包袱,构建高效网络

32 | 同源策略:为什么 XMLHttpRequest不能跨域请求资源?

33丨跨站脚本攻击(XSS):为什么 Cookie中有HttpOnly属性?

34 | CSRF攻击: 陌生链接不要随便

35 | 安全沙箱:页面和系统之间的隔离墙

36 | HTTPS: 让数据传输更安全

00开篇词丨参透了浏览器的工作原理, 你就能解决80%的前端难题

01 | Chrome架构: 仅仅打开了1个 页面,为什么有4个进程?

02 | TCP协议:如何保证页面文件能被完整送达浏览器?

03 | HTTP请求流程: 为什么很多站 点第二次打开速度会很快?

04 | 导航流程: 从输入URL到页面展示, 这中间发生了什么?

05 | 渲染流程(上):HTML、CSS 和JavaScript,是如何变成页面的?

06 | 渲染流程(下):HTML、CSS 和JavaScript,是如何变成页面的?

07 | 变量提升: JavaScript代码是按顺序执行的吗?

08 | 调用栈:为什么JavaScript代码会出现栈溢出?

09 | 块级作用域: var缺陷以及为什 - 么要引入Tet和const?

10 | 作用域链和闭包:代码中出现相同的变量,JavaScript引擎是如何选择的?

11 | this: 从JavaScript执行上下文的视角讲清楚this

12 | 栈空间和堆空间:数据是如何存储的?

13 | 垃圾回收: 垃圾数据是如何自动回收的?

14 | 编译器和解释器: V8是如何执 行一段JavaScript代码的?

15 | 消息队列和事件循环:页面是怎么"活"起来的?

16 | WebAPI: setTimeout是如何实现的?

17 | WebAPI: XMLHttpRequest是 怎么实现的?

18 | 宏任务和微任务: 不是所有任 务都是一个待遇 加餐三|加载阶段性能:使用Audits 来优化Web性能

结束语 | 大道至简

加餐

加餐一 | 浏览上下文组:如何计算 Chrome中渲染进程的个数?

加餐二 | 任务调度:有了 setTimeOut,为什么还要使用rAF?