

# 前端复习课



v1.0.0

微信关注：前端复习课，持续更新

# 前端复习课

## 1-JAVASCRIPT

### 内置类型

- 介绍一下JS内置类型有哪些
  - JavaScript 数据类型和数据结构
- 介绍一下typeof区分类型的原理
  - JavaScript中typeof原理探究?
- 介绍一下类型转换
  - 深入理解JS的类型、值、类型转换

### 作用域

- 说说你对javascript的作用域的理解
  - JavaScript深入之词法作用域和动态作用域
- 什么是作用域链
  - JavaScript深入之作用域链
- 解释下 let 和 const 的块级作用域
  - ES6块级作用域

### 执行上下文

- 说说你对执行上下文的理解
  - JavaScript深入之执行上下文栈
  - 深入理解JavaScript执行上下文和执行栈

### 闭包

- 对闭包的看法，为什么要用闭包？说一下闭包原理以及应用场景
  - JavaScript闭包的底层运行机制
  - 对闭包的看法，为什么要用闭包？说一下闭包原理以及应用场景
  - 发现 JavaScript 中闭包的强大威力
- 闭包问题及优化
  - 闭包问题及解决
  - 为什么闭包不会被垃圾回收清除

## this指向

- 如何确定this指向
  - 彻底搞懂js this指向问题
- 改变 this 指向的方式都有哪些
- 介绍箭头函数的 this
  - 什么时候你不能使用箭头函数?

## 原型/继承

- 说一下对原型链的理解，画一个经典的原型链图示
  - 进阶必读：深入理解 JavaScript 原型
  - 最详尽的 JS 原型与原型链终极详解，没有「可能是」。 (一)
  - 最详尽的 JS 原型与原型链终极详解，没有「可能是」。 (二)
  - 最详尽的 JS 原型与原型链终极详解，没有「可能是」。 (三)
- 举例说明js如何实现继承
  - 做完这48道题彻底弄懂JS继承
- ES5/ES6 的继承除了写法以外还有什么区别
  - ES5/ES6 的继承除了写法以外还有什么区别?

## 事件循环

- 你对事件循环有了解吗？说说看！
  - Event Loop的规范和实现
  - 彻底吃透 JavaScript 执行机制
  - 前端中的事件循环eventloop机制
- 微任务和宏任务有什么区别
- 浏览器和Node 事件循环的区别
  - 浏览器与Node的事件循环(Event Loop)有何区别

## 异步编程

- 异步解决方案有哪些
  - 还在找什么，JavaScript的异步编程解决方案全在这里了
- Promise
  - 实现一个Promise
  - 实现Promise.all, race, allSeleted等
  - async/await

## 2-DOM

### 事件

- 事件冒泡、捕获等理解
  - 你真的理解 事件冒泡 和 事件捕获 吗?
- 介绍下浏览器事件委托
  - 事件委托
- 实现一个自定义事件
  - 创建自定义事件

### dom操作

- 原生JS DOM操作方法汇总

### 位置与大小

- JavaScript获取DOM元素位置和尺寸大小
  - JavaScript获取DOM元素位置和尺寸大小

## 3-CSS

### BFC

- BFC 是什么? 触发 BFC 的条件是什么? 有哪些应用场景?
  - 前端面试之BFC
  - 怎么回答当面试官问你什么是BFC

### 1px

- 如何解决移动端 Retina 屏 1px 像素问题
  - 移动端1px解决方案

### position

- position 有哪些属性
- position: sticky 用过没, 有啥效果
  - 杀了个回马枪, 还是说说position:sticky吧

## flex

- flex:1 的完整写法是？分别是什么意思？
  - [Flex 布局教程：语法篇](#)

## 重绘回流

- 介绍下重绘和回流（Repaint & Reflow），以及如何优化
  - [介绍下重绘和回流（Repaint & Reflow），以及如何优化](#)
- 说说你对GPU的理解，举例说明哪些元素能触发GPU硬件加速？
  - [Web 性能优化-CSS3 硬件加速\(GPU 加速\)](#)

## 常见布局

- 居中
  - [面试官：你能实现多少种水平垂直居中的布局（定宽高和不定宽高）](#)
- 单列布局
- 两列自适应布局
- 三栏布局
- 粘连布局
  - [几种常见的CSS布局](#)

## 动画实现

- css 如何实现动画
  - [CSS3动画实践](#)
  - [CSS动画的性能优化](#)
- 使用纯CSS代码实现动画的暂停与播放
  - [纯 CSS 方式实现 CSS 动画的暂停与播放！](#)
- 使用css3动画代替js的动画有什么好处？
  - [使用CSS3动画代替JS动画的好处](#)

## 盒模型

- [CSS盒模型详解（图文教程）](#)

# 4-浏览器

## 跨域

- 跨域通信有哪些方式
  - 前端跨域通信的几种方式
  - CORS 简单请求+预检请求（彻底理解跨域）
- 表单可以跨域吗
  - 为什么form表单提交没有跨域问题，但ajax提交有跨域问题？

## 从输入URL到页面展示，这中间发生了什么？

- 一文摸透从输入URL到页面渲染的过程
- 从输入 URL 到页面展示到底发生了什么？看完吊打面试官！
- 从输入 URL 到页面展示，这中间发生了什么？
- 从输入URL到页面展示，这中间发生了什么？
- 前端经典面试题: 从输入URL到页面加载发生了什么？

## HTML、CSS和JavaScript，是如何变成页面的？

- 浏览器的工作原理
- 从 8 道面试题看浏览器渲染过程与性能优化

## chrome仅仅打开了1个页面，为什么有4个进程？

- 仅仅打开了1个页面，为什么有4个进程？
- 仅仅打开了1个页面，为什么有4个进程？

## localStorage

- 关于客户端存储的前端面试题总结

## cookie

- 聊一聊cookie

# 5-网络

## HTTP

- 你知道哪些http头部
  - 详解 HTTP 头部信息
- 说一下 Http 缓存策略，有什么区别，分别解决了什么问题
  - 浏览器缓存知识小结及应用
  - 浅谈 Web 缓存

## TCP

- 请描述处TCP的三次握手和四次挥手
  - 面试官，不要再问我三次握手和四次挥手
  - 两张动图-彻底明白TCP的三次握手与四次挥手
- 为什么浏览器要限制tcp的连接最大个数？
  - 浏览器允许的并发请求资源数是什么意思？

## HTTP2

- HTTP2.0 特点
  - HTTP/2 相比 1.0 有哪些重大改进？
- 说一下 HTTP2.0 多路复用原理，多路复用有哪些优势？
  - HTTP 2.0 的二进制帧、流、多路复用

## HTTPS

- 简述https原理，以及与http的区别
  - 一个故事讲完https
  - 一次安全可靠的通信——HTTPS原理
  - HTTPS系列2——证书的信任链校验：certificate trust chain

## CDN

- CDN 是什么？描述下 CDN 原理？为什么要用 CDN？
  - 关于 cdn、回源等问题一网打尽

## DNS

- DNS 查询的过程，分为哪两种，是怎么一个过程
  - 浅析DNS域名解析过程

# 6-框架

## vue

- 你知道Vue响应式数据原理吗？Proxy 与 Object.defineProperty 优劣对比
  - 深入 Vue 响应式原理，活捉一个 MVVM
  - Vue3 的响应式和以前有什么区别，Proxy 无敌？
  - 面试官：实现双向绑定Proxy比defineproperty优劣如何？
  - Vue3 中的数据侦测
- Vue2.x组件通信有哪些方式
  - vue中8种组件通信方式
- Vue 中的 computed 和 watch 的区别在哪里
  - Vue 里的 computed 和 watch 的区别

- 组件中的data为什么是一个函数
  - Vue-组件的data属性为什么必须是函数?
  - 组件中的data为什么是一个函数而不是一个对象
- nextTick的实现原理是什么
  - Vue.nextTick, 了解一下?
- 说说你对keep-alive组件的了解
  - Vue keep-alive深入理解及实践总结
- 你都做过哪些Vue的性能优化
  - Vue 项目性能优化 — 实践指南
- vue3.0介绍
  - Vue3新特性一篇搞懂
- Composition API 的出现带来哪些新的开发体验, 为啥需要这个
  - Vue3快速上手指南-CompositionAPI
- vuex
  - 什么情况下使用 Vuex
    - 什么时候该用vuex?
  - 可以直接修改state的值么
    - vuex直接修改state 与 用dispatch / commit来修改state的差异
  - 为什么 Vuex 的 mutation 中不能做异步操作
    - vuex中为什么把异步操作封装在action, 把同步操作放在mutations?
  - v-model和vuex有冲突么
    - 表单处理
- router
  - 路由懒加载是什么意思? 如何实现路由懒加载
    - 路由懒加载
  - Vue-router 导航守卫有哪些
    - 导航守卫
  - vue路由hash模式和history模式实现原理分别是什么, 他们的区别是什么
    - 前端路由简介以及vue-router实现原理

## react

- React 事件绑定原理
  - 深入理解React: 事件机制原理
- React中的setState缺点是什么呢
- React组件通信如何实现
  - React 组件通信的五种方式
- 类组件和函数组件的区别
  - React 函数式组件和类组件的区别, 不是只有state和性能!
- 请你说说React的路由是什么



- 从路由原理出发，深入阅读理解react-router 4.0的源码
- React有哪些性能优化的手段
  - 21个React性能优化技巧
  - React性能优化的8种方式了解一下?
- React hooks用过吗，为什么要用
  - 为什么需要React Hooks?
  - 一文彻底搞懂react hooks的原理和实现

## vue/react

- 虚拟DOM的优劣如何
  - 虚拟 DOM 到底是什么?
  - 虚拟DOM的优劣如何
- v-for循环中key有什么作用?
  - Vue2.0 v-for 中 :key 到底有什么用?
  - 写 React / Vue 项目时为什么要在列表组件中写 key，其作用是什么?
- React 和 Vue 的 diff 时间复杂度从  $O(n^3)$  优化到  $O(n)$ ，那么  $O(n^3)$  和  $O(n)$  是如何计算出来的
  - React 和 Vue 的 diff 时间复杂度从  $O(n^3)$  优化到  $O(n)$ ，那么  $O(n^3)$  和  $O(n)$  是如何计算出来的?
  - React 和 Vue 的 diff 时间复杂度从  $O(n^3)$  优化到  $O(n)$ ，那么  $O(n^3)$  和  $O(n)$  是如何计算出来的?
- 聊聊 Redux 和 Vuex 的设计思想
  - Vuex、Flux、Redux、Redux-saga、Dva、MobX

## 7-工程化

### 脚手架

- 你知道什么是脚手架吗?
  - 前端脚手架，听起来玄乎，实际呢?
  - 自制前端脚手架
- 你们公司有自己的脚手架工具么，他是怎么工作的

### 构建工具

- webpack
  - webpack的核心思想是什么
    - Webpack原理浅析
    - 理解webpack原理，手写一个100行的webpack
  - Loader和Plugin的区别
    - webpack 中 loader 和 plugin 的区别是什么
  - 有哪些常见的Loader和Plugin，简单聊一聊各自的作用
    - Webpack 入门以及常见 Loader 和常用插件的使用

- 说一下 Webpack 的热更新原理吧
  - [Webpack HMR 原理解析](#)
- 如何优化 Webpack 的构建速度
  - [优化 Webpack 的构建速度](#)
- 自己写过Loader和Plugin么
  - [webpack loader和plugin编写](#)
  - [Writing a Loader](#)
- 代码分割的本质是什么？有什么意义呢
  - [代码分离](#)
  - [详解webpack code splitting](#)
- 说下 tree-shaking 的原理
  - [Webpack Tree shaking 深入探究](#)
- babel原理
  - [一文彻底读懂Babel](#)

## 项目部署

- 你们公司项目发布流程是什么样的
- 前端资源发布路径怎么实现非覆盖式发布（平滑升级）？
  - [前端开发体系建设日记](#)
- SSR项目是如何发布的

## 内部包

- 你有发布过自己的npm包吗？流程是怎样的？
- 介绍下 npm 模块安装机制，为什么输入 npm install 就可以自动安装对应的模块？
  - [每一个前端工程师都应该掌握的 npm install 原理](#)
- 你会搭建私有的npm仓库吗？怎么搭建？
  - [分分钟教会你搭建企业级的 npm 私有仓库](#)

## 开发规范

- eslint

## 运维

- nginx
- cdn

## git

## 构建优化

- Webpack 为什么慢，如何进行优化
  - 优化 Webpack 的构建速度

# 8-性能

## 页面是否可以快速加载

- 网络优化
  - http2
  - CDN
- 代码大小
  - 代码分割
  - 代码压缩
  - tree-shaking
- SSR
  - 彻底理解服务端渲染 - SSR原理
- 离线
  - h5 秒开方案大全
  - Hybrid App 离线包方案实践（附开源代码）
  - 极致的 Hybrid：航旅离线包再加速！
- 缓存
  - 浏览器缓存知识小结及应用
  - 浅谈 Web 缓存

## 是否允许用户快速开始与之交互

- 渲染优化
  - 从 8 道面试题看浏览器渲染过程与性能优化
- web worker
  - 使用Web Worker优化代码

## 滚动和动画是否流畅

- 如何优化大数据列表（10万+）的性能？说说你的方案
  - 「前端进阶」高性能渲染十万条数据(虚拟列表)
- requestAnimationFrame
- 动画性能如何检测
- js动画和css动画有什么区别
- 分离图层做动画有什么好处呢
  - css3动画优化

- 重绘重排
  - Web 性能优化-CSS3 硬件加速(GPU 加速)

## 图片优化

- 图片压缩
- 图片懒加载
  - 图片懒加载从简单到复杂

## 骨架屏+合理的loading

- 网页骨架屏自动生成方案 (dps)
- 一种自动化生成骨架屏的方案

# 9-监控

## 异常

- 项目中如何进行异常捕获
  - 前端开发不得不知道的异常捕获技巧
- 为什么会有那么多Script Error错误
  - Script error.全面解析

## 性能

- 前端是如何监控性能的
  - 腾讯前端团队是如何做web性能监控的?
  - 蚂蚁金服如何把前端性能监控做到极致?

## 埋点

- 页面埋点怎么实现
  - web 埋点实现原理了解一下

为什么通常在发送数据埋点请求的时候使用的是 1x1 像素的透明 gif 图片

- 为什么通常在发送数据埋点请求的时候使用的是 1x1 像素的透明 gif 图片?

sendbeacon

# 10-安全

## 2020全网最全前端安全综述

### XSS

- [如何防止XSS攻击?](#)

### csrf

- [如何防止CSRF攻击?](#)

### 网络传输安全

- [一次安全可靠的通信——HTTPS原理](#)[000046a5fdc7802a15f7508b556413](#)
- [HTTPS中间人攻击实践（原理·实践）](#)

### 接口加签

- [API 接口的参数加密签名设计思路（参数加密+超时处理+私钥验证+Https）](#)

### 接口加密

### 接口防重放

- [Api 接口安全-防篡改,防重放理解总结](#)

### 环境检测

### 代码加密混淆

### 无处不在的验证码s

- [验证码WEB端产品调研（一）：Google reCAPTCHA](#)
- [验证码WEB端产品调研（二）：极限验证](#)

### 浏览器为什么要阻止跨域请求？如何解决跨域？每次跨域请求都需要到达服务端吗？

- [为什么浏览器要限制跨域访问？](#)

# 11-NODE

### node 事件循环

- [Node.js 事件循环-比官方更全面](#)

谈谈 node 的内存泄漏

- 如何分析 Node.js 中的内存泄漏

node 中间层怎样做的请求合并转发

pm2 怎么做进程管理，进程挂掉怎么处理

SSR

- SSR解决了什么问题？有做过SSR吗？你是怎么做的？
  - 彻底理解服务端渲染 - SSR原理
- SSR这么好为啥不大规模使用了，基本公司项目还走的是前端渲染

GraphQL

- 使用过GraphQL么，解决了什么问题
- 和普通接口相比GraphQL有什么优势呢
  - GraphQL-BFF：微服务背景下的前后端数据交互方案

## 12-跨端

Hybrid App

- 说一下H5和native之间是如何进行交互的，知道jsbridge么
  - 小白必看，JSBridge 初探
  - Hybrid App技术解析 -- 原理篇
  - Hybrid App技术解析 -- 实战篇

Weex

- weex实现大致原理
  - 深入理解weex内核原理

RN

Flutter

## 13-微信开发

## 公众号

- 微信授权流程
  - 微信扫码登录的几秒钟里，到底发生了什么

## 小程序

- 了解微信小程序的底层实现吗
  - 小程序简介
- h5跟小程序有什么区别
- 五花八门的小程序框架实现原理知道么
  - 为何我们要用 React 来写小程序 - Taro 诞生记

# 14-新主题

## 微前端

- 你了解什么是微前端吗？说说你对它的理解？
  - 可能是你见过最完善的微前端解决方案

## serverless

- 说说你对serverless的理解，它对前端有什么影响？
  - Serverless 掀起新的前端技术变革

## 边缘计算

- 你听过边缘计算吗？说说你对它的理解？
  - 前端性能优化：当页面渲染遇上边缘计算
  - 什么是边缘计算？一文读尽秒懂

## WebAssembly

# 15-手写

## Promise

- 手写一个Promise
  - 面试官：“你能手写一个 Promise 吗”
- 使用Promise实现红绿灯交替重复亮
  - 使用Promise实现红绿灯交替重复亮
- Promise.all, race, allSettled

- sendRequests(urls, max, callback), 同一时间最多发起max个请求, 请求全部完成后执行callback
- maxRequest(fn, maxNum), 实现maxRequest, 成功后resolve结果, 失败后重试, 尝试超过一定次数才真正的reject
- 页面上有一个输入框, 两个按钮, A按钮和B按钮, 点击A或者B分别会发送一个异步请求, 请求完成后, 结果会显示在输入框中。用户随机点击A和B多次, 要求输出显示结果时, 按照用户点击的顺序显示
  - 一道赋值面试题引发的思考3【并发数控制】

## this

- call/apply
- bind
  - 手写call、apply、bind实现及详解
- new
  - new操作符的实现
- 链式调用4.add(1).add(2).add(3)

## 原型链

- instanceof
  - 手写一个instanceOf
- 各种继承
  - JavaScript深入之继承的多种方式和优缺点

## 闭包

- 修改函数正确执行
- 柯里化
  - JavaScript专题之函数柯里化

## 防抖节流

- debounce
  - JavaScript专题之跟着underscore学防抖
- throttle
  - JavaScript专题之跟着 underscore 学节流

## 网络请求

- 手写AJAX
  - 手写 AJAX
- 手写JSONP
  - jsonp的原理与实现



## 设计模式

- EventEmitter
  - 从观察者模式到手写EventEmitter源码
- singleton

## 深拷贝

- JS基本数据类型和引用数据类型的区别及深浅拷贝
- JavaScript专题之深浅拷贝

# 16-算法

## 字符串

- 反转字符串
- 字符串中的第一个唯一字符
- 最长公共前缀

## 数组

- 合并两个有序数组
- 数组中重复的数字
- 两个数组的交集
- 旋转数组
- 两数之和

## 链表

- 合并两个有序链表
- 删除中间节点
- 移除链表元素
- 反转链表
- 删除链表的倒数第N个节点
- 环形链表
- 从尾到头打印链表

## 二叉树

- 二叉树的遍历-前序中序后序层次
- 二叉树的最大深度
- 二叉搜索树
- BST 的查找
- 平衡二叉树
- 完全二叉树

## 栈/队列

- 有效的括号

## 排序

- 冒泡排序
- 选择排序
- 插入排序
- 希尔排序
- 归并排序
- 快速排序
- 堆排序

## 递归

- 斐波那契数列
- 汉诺塔问题

## 二分法

- 爱吃香蕉的珂珂
- x的平方根
- 寻找两个正序数组的中位数
- 搜索旋转排序数组
- 搜索插入位置

## 动态规划

- 最长上升子序列
- 爬楼梯
- 最大子序和
- 最小路径和
- 打家劫舍
- 买卖股票的最佳时机

## 贪心与分治

- 分发饼干
- 柠檬水找零
- 模拟行走机器人

## 滑动窗口

- 滑动窗口最大值
- 无重复字符的最长子串
- 盛最多水的容器

## 位运算

- N皇后 II
- 颠倒二进制位
- 位1的个数

## 17-项目

项目开发中有遇到什么挑战没？

对哪个项目印象比较深刻深刻，遇到最难的项目是啥？

项目研发流程中作为前端开发一般扮演的啥角色？

现在有的项目中觉得哪些项目可以继续优化，为啥没有优化？

平时写项目总结么，一般总结哪些东西？

## 18-技术之外

工作中能够持续学习么？

学习的动力怎么来的，如何维持？

未来会有什么样的规划？

对于加班你是怎么看的？

说下你学习前端的历程吧？

前端未来展望？



微信关注：前端复习课，持续更新