# 学习网站

<https://www.w3school.com.cn/>

[CSS 网格布局 | 菜鸟教程 (runoob.com)](https://www.runoob.com/css3/css-grid.html)

[万字总结我在寒冬里的面试准备经历 - 掘金 (juejin.cn)](https://juejin.cn/post/7270095064440864804)

[前端面试题汇总 (yuque.com)](https://www.yuque.com/cuggz/interview)

[6分钟彻底掌握vue的diff算法，前端面试不再怕！\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1JR4y1R7Ln/?spm_id_from=333.337.search-card.all.click&vd_source=49e0869b12b6f504cfdd91857a38d4d1)

[Code Hot](https://codehot.cn/list/os)

[小林coding (xiaolincoding.com)](https://xiaolincoding.com/)

# 一、HTML

## src与href

[href和src的区别\_href和src的区别是什么-CSDN博客](https://blog.csdn.net/weixin_44063225/article/details/111846325?ops_request_misc=&request_id=&biz_id=102&utm_term=src%E5%92%8Chref&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-5-111846325.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## defer和async

[async和defer的区别?-CSDN博客](https://blog.csdn.net/qq_44621394/article/details/126545273?ops_request_misc=%7B%22request%5Fid%22%3A%22171566949316800182169824%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171566949316800182169824&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-1-126545273-null-null.142^v100^pc_search_result_base6&utm_term=defer%E5%92%8Casync%E7%9A%84%E5%8C%BA%E5%88%AB&spm=1018.2226.3001.4187)

# 二、CSS

## px、em、rem

[px、rem、em的区别与联系\_px rem-CSDN博客](https://blog.csdn.net/weixin_44019523/article/details/114155763?ops_request_misc=&request_id=&biz_id=102&utm_term=px%E3%80%81em%E3%80%81rem%E7%9A%84%E5%8C%BA%E5%88%AB&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-114155763.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## 关于BFC

[对BFC的理解以及如何创建BFC\_对bfc的理解,如何创建bfc-CSDN博客](https://blog.csdn.net/Miller777_/article/details/136495992?ops_request_misc=%7B%22request%5Fid%22%3A%22171438168116800211523637%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171438168116800211523637&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-136495992-null-null.142^v100^pc_search_result_base6&utm_term=%E5%AF%B9BFC%E7%9A%84%E7%90%86%E8%A7%A3%EF%BC%8C%E5%A6%82%E4%BD%95%E5%88%9B%E5%BB%BABFC&spm=1018.2226.3001.4187)

## 清除浮动

[css之clear属性，both left right详解，解决父元素高度塌陷-CSDN博客](https://blog.csdn.net/qq_42667613/article/details/123478712?ops_request_misc=%7B%22request%5Fid%22%3A%22171202347716800182743268%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171202347716800182743268&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-7-123478712-null-null.142^v100^pc_search_result_base6&utm_term=clear both%E5%8E%9F%E7%90%86&spm=1018.2226.3001.4187)

## Flex布局

[一文看懂flex布局\_flex-wrap: wrap;-CSDN博客](https://blog.csdn.net/qq_42825643/article/details/124237261?ops_request_misc=%7B%22request%5Fid%22%3A%22171409913416800182116128%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171409913416800182116128&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-124237261-null-null.142^v100^pc_search_result_base6&utm_term=flex-wrap&spm=1018.2226.3001.4187)

[flex弹性布局教程-12容器属性align-content\_align-content: flex-start;-CSDN博客](https://blog.csdn.net/chenjiebin/article/details/120517726?ops_request_misc=&request_id=&biz_id=102&utm_term=flex%E5%B8%83%E5%B1%80align-content&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-2-120517726.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## Flex 1

[你真的知道flex: 1；是什么意思吗？\_flex:1属性是什么意思-CSDN博客](https://blog.csdn.net/m0_58875326/article/details/124444419?ops_request_misc=%7B%22request%5Fid%22%3A%22171567863316800182123983%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171567863316800182123983&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-1-124444419-null-null.142^v100^pc_search_result_base6&utm_term=flex 1&spm=1018.2226.3001.4187)

## Grid布局

[css 网格布局 grid 详解\_grid: max-content 1fr / none 是什么意思-CSDN博客](https://blog.csdn.net/2301_76669854/article/details/134758040?ops_request_misc=%7B%22request%5Fid%22%3A%22171410218116800197043577%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171410218116800197043577&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-6-134758040-null-null.142^v100^pc_search_result_base6&utm_term=css%E7%BD%91%E6%A0%BC%E5%B8%83%E5%B1%80grid&spm=1018.2226.3001.4187)

## 元素居中的方案

[css实现元素居中的6种方法\_css居中-CSDN博客](https://blog.csdn.net/chaoPerson/article/details/126987151?ops_request_misc=&request_id=&biz_id=102&utm_term=css %E5%B1%85%E4%B8%AD&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-6-126987151.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

# 三、JS

## 数据类型

[js数据类型有哪些\_js bigint和symbol属于基本数据类型麽-CSDN博客](https://blog.csdn.net/qq_34402069/article/details/131164717?ops_request_misc=%7B%22request%5Fid%22%3A%22171444542416800197074299%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171444542416800197074299&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-131164717-null-null.142^v100^pc_search_result_base6&utm_term=js%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B&spm=1018.2226.3001.4187)

## let const var

 注意：var只有在全局作用域里声明时才会添加全局属性，如果声明变量没用任何关键字，那么不管在哪声明则都会成为全局属性

## commonjs与es6模块化有什么区别

[面试题57：commonjs与es6模块化有什么区别\_commonjs与es模块化的区别-CSDN博客](https://blog.csdn.net/qq_51066068/article/details/125579114?ops_request_misc=%7B%22request%5Fid%22%3A%22171513486716800211585812%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171513486716800211585812&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-18-125579114-null-null.142^v100^pc_search_result_base1&utm_term= ES6%E6%A8%A1%E5%9D%97%E4%B8%8ECommonJS%E6%A8%A1%E5%9D%97%E6%9C%89%E4%BB%80%E4%B9%88%E5%BC%82%E5%90%8C&spm=1018.2226.3001.4187)

## Object.is() 与比较操作符 “ ===” 、 “ ==” 的区别

[Object.is() 与比较操作符 “ ===” 、 “ ==” 的区别-CSDN博客](https://blog.csdn.net/Cshaosun/article/details/137968025?ops_request_misc=&request_id=&biz_id=102&utm_term=  === == Object.is()&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-9-137968025.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## 深浅拷贝问题

[js中的深拷贝与浅拷贝\_js 深拷贝-CSDN博客](https://blog.csdn.net/m0_61480985/article/details/128192967?ops_request_misc=%7B%22request%5Fid%22%3A%22171446012616800222829542%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171446012616800222829542&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-128192967-null-null.142^v100^pc_search_result_base6&utm_term=js ...%E6%B7%B1%E6%8B%B7%E8%B4%9D%E8%BF%98%E6%98%AF%E6%B5%85%E6%8B%B7%E8%B4%9D&spm=1018.2226.3001.4187)

## Proxy和defineproperty

[defineProperty和proxy区别\_proxy和definepropery的区别-CSDN博客](https://blog.csdn.net/weixin_43443341/article/details/124041094)

[Proxy 与 defineProperty 的理解、区别、优势、劣势\_proxy和defineproperty的优劣对比-CSDN博客](https://blog.csdn.net/qq_38290251/article/details/135280017?ops_request_misc=&request_id=&biz_id=102&utm_term=Proxy%E4%B8%8Edefineproperty&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-135280017.142^v100^control&spm=1018.2226.3001.4187)

## Promise、async\await

[Promise详解大全：介绍、九个方法使用和区别、返回值详解\_new promise 返回什么-CSDN博客](https://blog.csdn.net/qq_53669554/article/details/131598219?ops_request_misc=&request_id=&biz_id=102&utm_term=promise then%E7%9A%84%E8%BF%94%E5%9B%9E%E5%80%BC&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-8-131598219.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

then和catch的返回值

[promise的三种状态及.then() .catch() .finally() .all() .race()的使用\_.then .catch-CSDN博客](https://blog.csdn.net/wxiao_xiao_miao/article/details/120374015?ops_request_misc=&request_id=&biz_id=102&utm_term=promise catch%E7%9A%84%E8%BF%94%E5%9B%9E%E5%80%BC&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-120374015.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

[Promise - async await的基本用法以及使用陷阱，高效使用技巧\_await promise-CSDN博客](https://blog.csdn.net/qfc_128220/article/details/121757898?ops_request_misc=&request_id=&biz_id=102&utm_term=promise async await%E4%BD%BF%E7%94%A8&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-121757898.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## promise、Generator、async/await它们之间的关系

[async/await、Generator、Promise之间的关联和区别\_async 和await和generator得关系-CSDN博客](https://blog.csdn.net/weixin_57837275/article/details/135144709?ops_request_misc=%7B%22request%5Fid%22%3A%22171575040516800226526136%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171575040516800226526136&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-135144709-null-null.142^v100^pc_search_result_base6&utm_term=promise%E3%80%81Generator%E3%80%81async/await%E5%AE%83%E4%BB%AC%E4%B9%8B%E9%97%B4%E7%9A%84%E5%85%B3%E7%B3%BB&spm=1018.2226.3001.4187)

promise如何解决回调地狱：

[promise、async,await、generator函数\_await promise generator-CSDN博客](https://blog.csdn.net/weixin_57748960/article/details/116501350?ops_request_misc=&request_id=&biz_id=102&utm_term=promise%E3%80%81Generator%E3%80%81async/await%E5%AE%83&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-3-116501350.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

Generator如何使异步任务同步：

[Generator与yield 语法糖 async和await\_async await yield-CSDN博客](https://blog.csdn.net/qq_41328247/article/details/108630826?ops_request_misc=%7B%22request%5Fid%22%3A%22171575429216800222882396%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171575429216800222882396&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-11-108630826-null-null.142^v100^pc_search_result_base6&utm_term=generator yield %E5%90%8E%E9%9D%A2promise&spm=1018.2226.3001.4187)

## Proxy

[JS中的Proxy代理详解\_js代理-CSDN博客](https://blog.csdn.net/m0_37394102/article/details/133124912?ops_request_misc=%7B%22request%5Fid%22%3A%22171392081416777224419017%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171392081416777224419017&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-133124912-null-null.142^v100^pc_search_result_base6&utm_term=js proxy&spm=1018.2226.3001.4187)

## Call,apply,bind

[js的call()、apply()、bind()解析\_js call bind allpy区别-CSDN博客](https://blog.csdn.net/qq_44308109/article/details/124928563?ops_request_misc=%7B%22request%5Fid%22%3A%22171394399116800188528635%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171394399116800188528635&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~baidu_landing_v2~default-6-124928563-null-null.142^v100^pc_search_result_base6&utm_term=js bind call apply&spm=1018.2226.3001.4187)

[JS高级——实现apply、call、bind函数\_js fn.apply-CSDN博客](https://blog.csdn.net/Yuanyuan__/article/details/130670123?ops_request_misc=&request_id=&biz_id=102&utm_term=js bind call apply&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-4-130670123.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## 继承的方式

[JS中的八种继承方法\_js 继承-CSDN博客](https://blog.csdn.net/weixin_70134200/article/details/131730945?ops_request_misc=&request_id=&biz_id=102&utm_term=javascript%E6%9E%84%E9%80%A0%E5%87%BD%E6%95%B0%E7%BB%A7%E6%89%BF&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-4-131730945.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

[JS继承的几种方式及优缺点\_构造函数继承的缺点-CSDN博客](https://blog.csdn.net/qq_56088882/article/details/125823829?ops_request_misc=&request_id=&biz_id=102&utm_term=javascript%E6%9E%84%E9%80%A0%E5%87%BD%E6%95%B0%E7%BB%A7%E6%89%BF&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-6-125823829.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## http请求的格式

[HTTP请求消息数据格式详解(请求头,请求行,请求体)\_请求头数据和请求体数据啥意思-CSDN博客](https://blog.csdn.net/qq_40121580/article/details/107349943?ops_request_misc=%7B%22request%5Fid%22%3A%22171274582516800197039641%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171274582516800197039641&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-3-107349943-null-null.142^v100^pc_search_result_base1&utm_term=%E8%AF%B7%E6%B1%82%E8%A1%8C%E8%AF%B7%E6%B1%82%E5%A4%B4%E5%92%8C%E6%B6%88%E6%81%AF%E4%BD%93&spm=1018.2226.3001.4187)

## Js中的内存管理

[「前端进阶」JS中的内存管理 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/490835144)

[js---js使用闭包是否会产生内存泄露及解决方案\_js闭包内存泄露解决-CSDN博客](https://blog.csdn.net/h18377528386/article/details/126712640?ops_request_misc=%7B%22request%5Fid%22%3A%22171334572616800213044323%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171334572616800213044323&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-126712640-null-null.142^v100^pc_search_result_base6&utm_term=j%E2%80%86s%E9%97%AD%E5%8C%85%E9%80%A0%E6%88%90%E5%86%85%E5%AD%98%E6%B3%84%E9%9C%B2&spm=1018.2226.3001.4187)

## es6新特性

[最全的—— ES6有哪些新特性？\_es6新特性-CSDN博客](https://blog.csdn.net/ZLJ_999/article/details/124122540?ops_request_misc=%7B%22request%5Fid%22%3A%22171385973516800185824514%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171385973516800185824514&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-2-124122540-null-null.142^v100^pc_search_result_base6&utm_term=es6&spm=1018.2226.3001.4187)

## 尾调用

[【ES6标准入门】JavaScript的函数尾调调优\_尾调用优化js-CSDN博客](https://blog.csdn.net/m0_56132701/article/details/133520447?ops_request_misc=&request_id=&biz_id=102&utm_term=%E5%B0%BE%E8%B0%83%E7%94%A8&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-133520447.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## axios

[axios详解以及完整封装方法\_axios封装-CSDN博客](https://blog.csdn.net/zrblue/article/details/135318200?ops_request_misc=&request_id=&biz_id=102&utm_term=axios&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-135318200.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## 严格模式

[js代码中“use strict” 是什么意思？ 使用它的区别是什么？\_.javascript 代码中的”use strict”;是什么意思 ? 使用它区别是什么?-CSDN博客](https://blog.csdn.net/m0_74265396/article/details/135562367?ops_request_misc=&request_id=&biz_id=102&utm_term= use strict%E6%98%AF%E4%BB%80%E4%B9%88%E6%84%8F%E6%80%9D&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-6-135562367.nonecase&spm=1018.2226.3001.4187)

## 事件冒泡和事件捕获

[解释事件冒泡和事件捕获的原理，并说明它们之间的区别与使用场景-CSDN博客](https://blog.csdn.net/weixin_53291256/article/details/131697929?ops_request_misc=&request_id=&biz_id=102&utm_term=%E4%BA%8B%E4%BB%B6%E5%86%92%E6%B3%A1%E5%92%8C%E4%BA%8B%E4%BB%B6%E6%8D%95%E8%8E%B7&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-131697929.nonecase&spm=1018.2226.3001.4187)

## 手写new

[手写new函数的详解-CSDN博客](https://blog.csdn.net/myname_Christina/article/details/126239863?ops_request_misc=%7B%22request%5Fid%22%3A%22171446142216800226587433%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171446142216800226587433&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-126239863-null-null.142^v100^pc_search_result_base6&utm_term=%E6%89%8B%E5%86%99 new&spm=1018.2226.3001.4187)

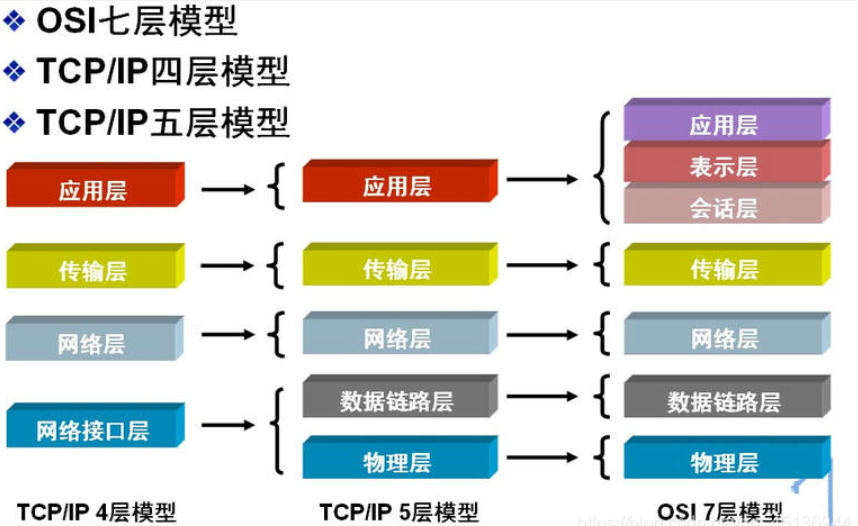
## webpack之loader与plugin

[webpack中loader和plugin的区别\_webpack loader和plugin区别-CSDN博客](https://blog.csdn.net/SH744/article/details/127423169?ops_request_misc=%7B%22request%5Fid%22%3A%22171522713316800222848162%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171522713316800222848162&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-5-127423169-null-null.142^v100^pc_search_result_base1&utm_term=webpackloader%E5%92%8Cplugin&spm=1018.2226.3001.4187)

## 跨域

# 四、网络

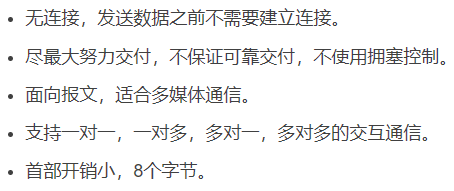
## 协议栈层次



## 各层协议有哪些

## UDP

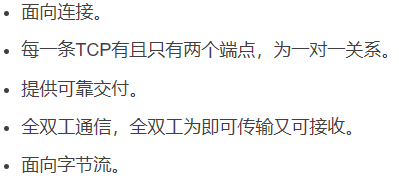
**适用于对网络通讯质量要求不高时，要求网络通讯速度要快的场景**

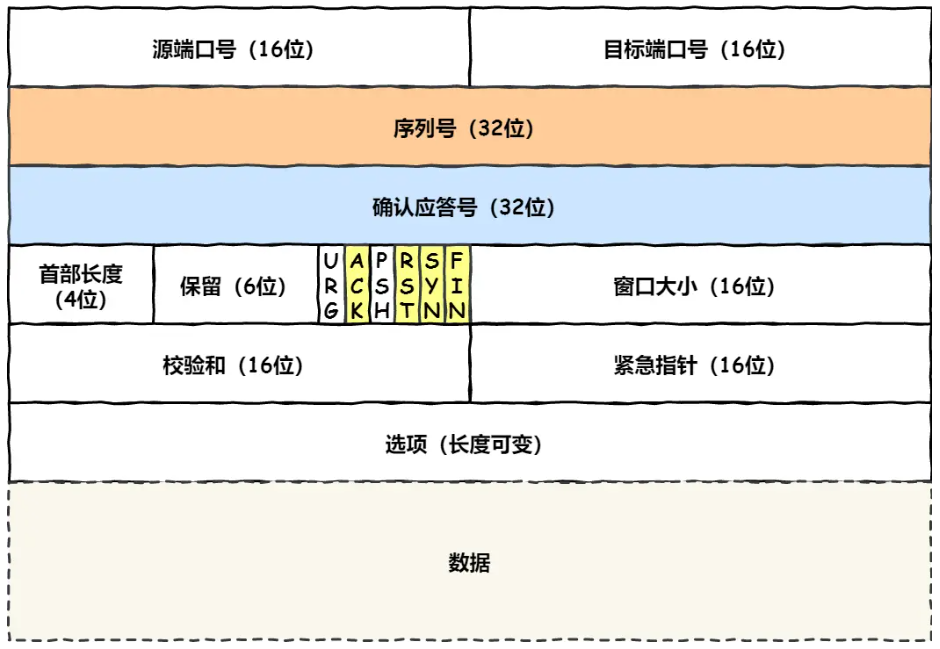




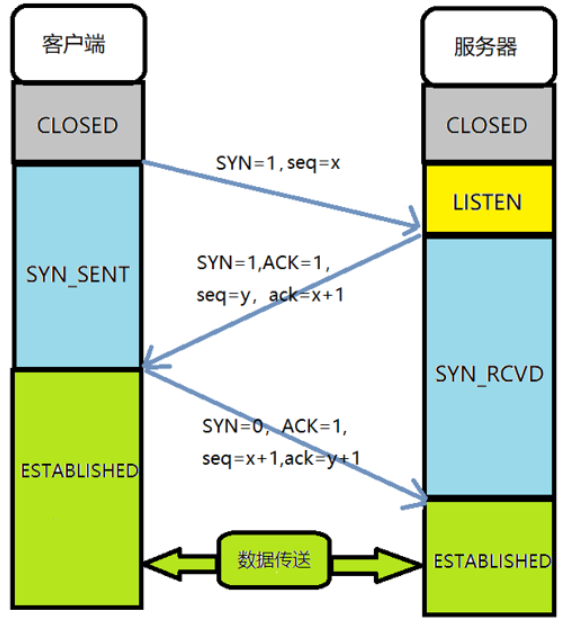
## TCP

**当对网络通讯质量有要求时，比如HTTP、HTTPS、FTP等传输文件的协议， POP3、SMTP等邮件传输的协议**





## 三次握手



## 为什么需要三次握手

三次握手才可以阻止重复历史连接的初始化（主要原因）

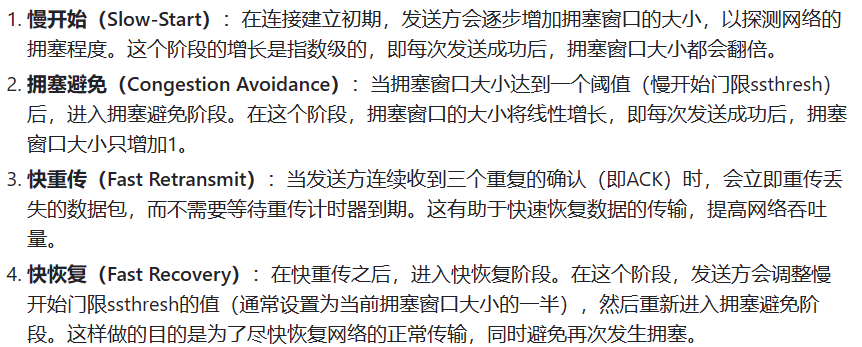
三次握手才可以同步双方的初始序列号

三次握手才可以避免资源浪费

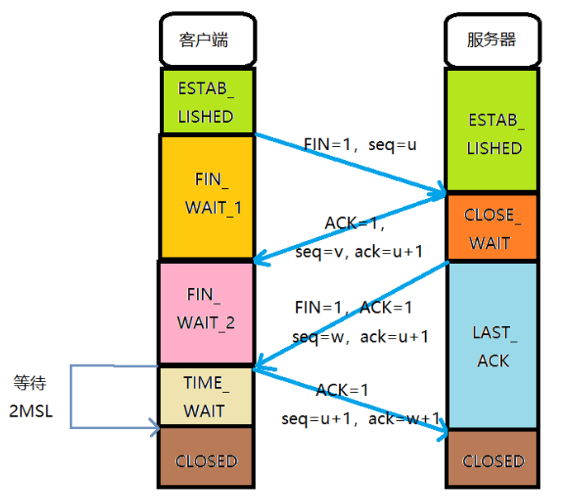
## TCP为什么可靠

[详细总结：TCP为何是可靠的\_tcp为什么是可靠连接-CSDN博客](https://blog.csdn.net/qq_40861091/article/details/102022443?ops_request_misc=%7B%22request%5Fid%22%3A%22171125457616800185868423%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171125457616800185868423&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-102022443-null-null.142^v99^pc_search_result_base1&utm_term=tcp%E4%B8%BA%E4%BB%80%E4%B9%88%E5%8F%AF%E9%9D%A0&spm=1018.2226.3001.4187)

## 拥塞控制



## 四次挥手

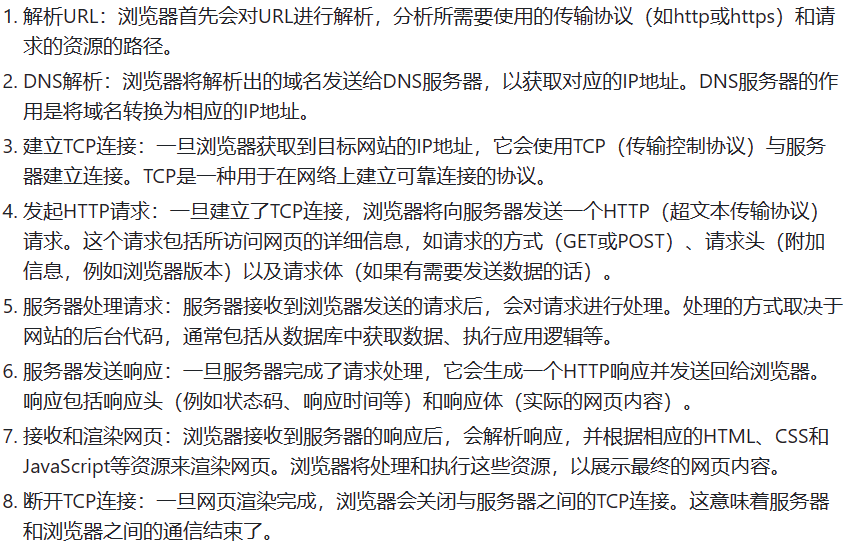


报文段最大生存时间MSL（Maximum SegmentLifetime）

## RST报文

[TCP重置报文段及RST常见场景分析-CSDN博客](https://blog.csdn.net/weixin_30670925/article/details/102234522?ops_request_misc=&request_id=&biz_id=102&utm_term=RST%E6%8A%A5%E6%96%87&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-3-102234522.142^v99^pc_search_result_base1&spm=1018.2226.3001.4187)

## 输入一个网址会发生什么



## DNS解析流程

[多张图带你彻底搞懂DNS域名解析过程\_域名解析过程以及缓存存放的位置-CSDN博客](https://blog.csdn.net/weixin_45629285/article/details/122969104?ops_request_misc=%7B%22request%5Fid%22%3A%22171133962516800197012486%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171133962516800197012486&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-122969104-null-null.142^v99^pc_search_result_base1&utm_term=dns%E5%9F%9F%E5%90%8D%E8%A7%A3%E6%9E%90%E8%BF%87%E7%A8%8B&spm=1018.2226.3001.4187)

## DNS劫持

## HTTP状态码



## GET和POST

GET 的语义是请求获取指定的资源。GET 方法是安全、幂等、可被缓存的。

POST 的语义是根据请求负荷（报文主体）对指定的资源做出处理，具体的处理方式视资源类型而不同。POST 不安全，不幂等，（大部分实现）不可缓存。

## 常见的http请求头以及响应头

[常见的http请求头以及响应头\_请求头和响应头的常见字段-CSDN博客](https://blog.csdn.net/hannah2233/article/details/125911821?ops_request_misc=%7B%22request%5Fid%22%3A%22171527036516800188575255%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171527036516800188575255&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-3-125911821-null-null.142^v100^pc_search_result_base1&utm_term=http%E8%AF%B7%E6%B1%82%E5%A4%B4&spm=1018.2226.3001.4187)

## http1.0、1.1、2.0、3.0

[HTTP 1.0 / 1.1 / 2.0 / 3.0 区别\_http/1.0-CSDN博客](https://blog.csdn.net/m0_52963553/article/details/129894192?ops_request_misc=&request_id=&biz_id=102&utm_term=http1.0 1.1 2.0 3.0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-7-129894192.142^v100^pc_search_result_base1&spm=1018.2226.3001.4187)

## HTTPS

## Session 和 cookie

# 五、Vue

## 双向绑定，如何双向更新数据

[Vue双向绑定：原理篇（详细）\_vue实现双向绑定原理-CSDN博客](https://blog.csdn.net/weixin_51670675/article/details/124069519?ops_request_misc=%7B%22request%5Fid%22%3A%22171195693616800197053191%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fblog.%22%7D&request_id=171195693616800197053191&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-7-124069519-null-null.nonecase&utm_term=vue%E5%8F%8C%E5%90%91%E7%BB%91%E5%AE%9A&spm=1018.2226.3001.4450)



## Vue生命周期

## Vue3 defineEmits组件自定义事件实现子=>父发送信号

[【超细节】Vue3组件事件怎么声明，defineEmits与emit\_vue3 defineemits-CSDN博客](https://blog.csdn.net/weixin_42373175/article/details/132088531?ops_request_misc=&request_id=&biz_id=102&utm_term=vue3 defineEmits(["submit"])&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-132088531.142^v100^pc_search_result_base1&spm=1018.2226.3001.4187)

## Vue3 defineExpose暴露子组件的属性和方法供父组件调用

[Vue3中的defineExpose\_vue3 defineexpose-CSDN博客](https://blog.csdn.net/weixin_59233142/article/details/135530798?ops_request_misc=&request_id=&biz_id=102&utm_term=vue3 defineExpose&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-4-135530798.nonecase&spm=1018.2226.3001.4187)

## Vue3 defineProps定义子组件的参数，父组件可以向子组件传参

## Vue3 v-model加在自定义组件上实现双向通信

[自定义组件V-Model-CSDN博客](https://blog.csdn.net/S_3405008677/article/details/126827852?ops_request_misc=%7B%22request%5Fid%22%3A%22171318945016800178537159%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171318945016800178537159&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-126827852-null-null.142^v100^pc_search_result_base1&utm_term=%E8%87%AA%E5%AE%9A%E4%B9%89%E7%BB%84%E4%BB%B6 v-model&spm=1018.2226.3001.4187)

[vue中内置指令v-model的作用和常见使用方法介绍以及在自定义组件上支持-CSDN博客](https://blog.csdn.net/shanghai597/article/details/134786130?ops_request_misc=%7B%22request%5Fid%22%3A%22171318945016800178537159%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171318945016800178537159&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~baidu_landing_v2~default-13-134786130-null-null.142^v100^pc_search_result_base1&utm_term=%E8%87%AA%E5%AE%9A%E4%B9%89%E7%BB%84%E4%BB%B6 v-model&spm=1018.2226.3001.4187)

## Vuex

[vuex和pina的区别\_pinia和vuex区别-CSDN博客](https://blog.csdn.net/qq_22182989/article/details/136445323?ops_request_misc=%7B%22request%5Fid%22%3A%22171525315716800182120769%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171525315716800182120769&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-136445323-null-null.142^v100^pc_search_result_base1&utm_term=vuex%E5%92%8Cpinia%E7%9A%84%E5%8C%BA%E5%88%AB%EF%BC%9F&spm=1018.2226.3001.4187)

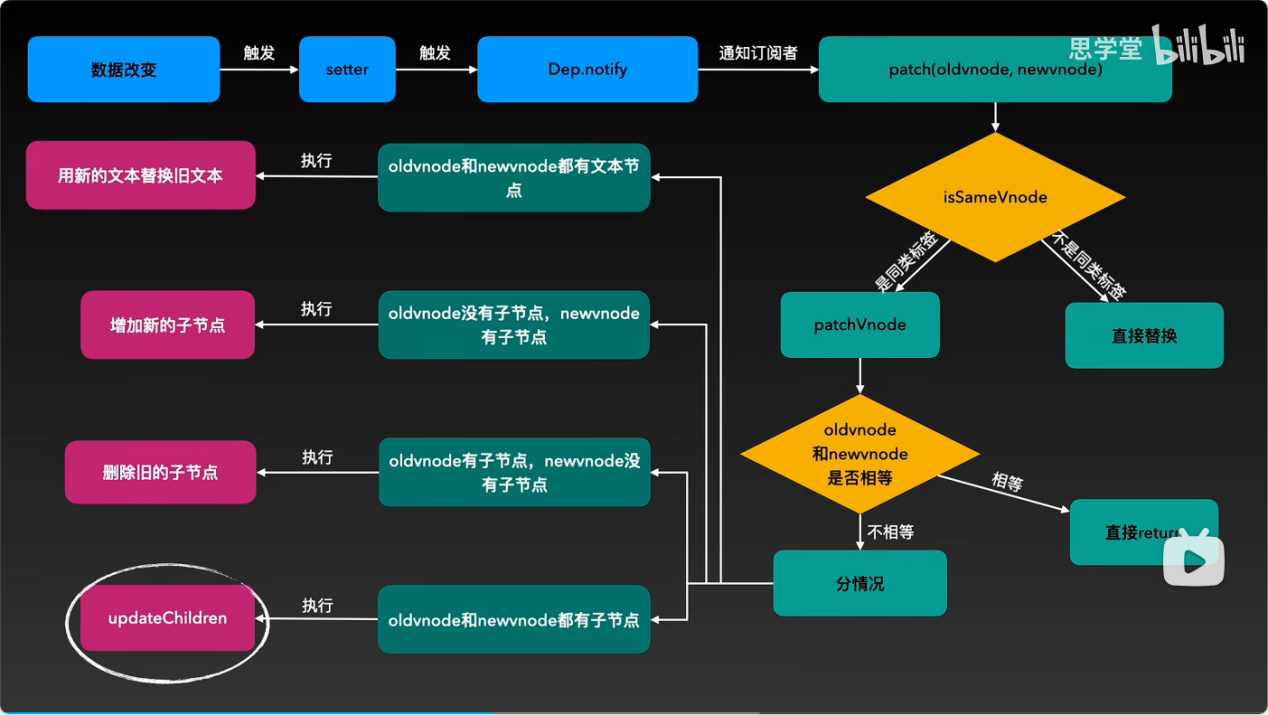
## Vue3和vue2的区别

[vue2和vue3的区别\_vue2和vue3区别-CSDN博客](https://blog.csdn.net/du_aitiantian/article/details/128902488?ops_request_misc=%7B%22request%5Fid%22%3A%22171525484816800188570912%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171525484816800188570912&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-128902488-null-null.142^v100^pc_search_result_base1&utm_term=Vue3%E5%92%8Cvue2%E7%9A%84%E5%8C%BA%E5%88%AB&spm=1018.2226.3001.4187)

## Diff算法

[vue 虚拟dom和diff算法详解\_vue的dom diff算法-CSDN博客](https://blog.csdn.net/weixin_42707287/article/details/113994483?ops_request_misc=%7B%22request%5Fid%22%3A%22171525493816800188588407%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171525493816800188588407&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-113994483-null-null.142^v100^pc_search_result_base1&utm_term=Diff%E7%AE%97%E6%B3%95&spm=1018.2226.3001.4187)

[【Vue】中Key的作用\_vue中key的作用-CSDN博客](https://blog.csdn.net/z914020826/article/details/127231981?ops_request_misc=&request_id=&biz_id=102&utm_term=vue key%E7%9A%84%E4%BD%9C%E7%94%A8&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-127231981.nonecase&spm=1018.2226.3001.4187)



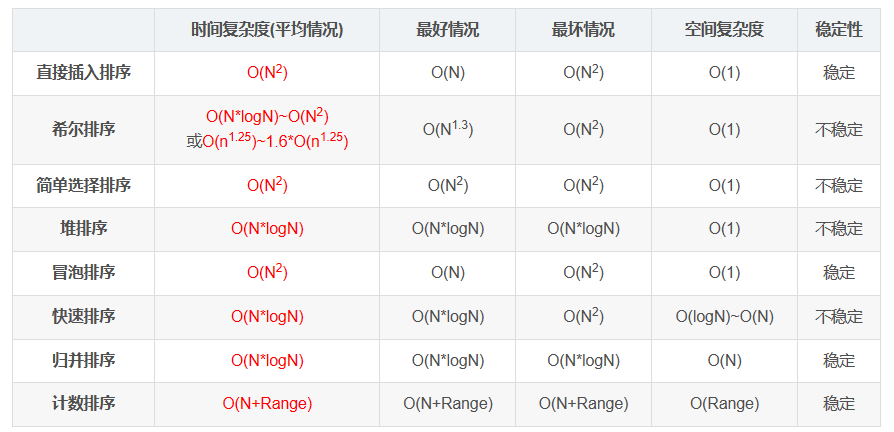
# 六、其他

## Session、cookie和JWT

[全网最细总结-Seesion,Cookie以及JWT的区别\_cookie,session,jwt-CSDN博客](https://blog.csdn.net/qq_42898642/article/details/131206223?spm=1001.2014.3001.5506)

## 排序算法

[万字总结——常见的八大排序算法（插入排序、希尔排序、选择排序、堆排序、冒泡排序、快速排序、归并排序、计数排序）\_排序算法总结-CSDN博客](https://blog.csdn.net/m0_73900674/article/details/132418128?ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%8E%92%E5%BA%8F%E7%AE%97%E6%B3%95&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-2-132418128.142^v100^pc_search_result_base1&spm=1018.2226.3001.4187)



## 免密登录过期怎么办

## 单例模式、中介者模式、状态模式在前端里的应用

[C++设计模式-状态模式详解\_状态模式c++-CSDN博客](https://blog.csdn.net/wb175208/article/details/85239414?ops_request_misc=%7B%22request%5Fid%22%3A%22171532689516800178599666%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171532689516800178599666&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~baidu_landing_v2~default-2-85239414-null-null.142^v100^pc_search_result_base1&utm_term=%E7%8A%B6%E6%80%81%E6%A8%A1%E5%BC%8FC++&spm=1018.2226.3001.4187)

## Keep-alive原理

[Vue中keep-alive缓存的详解（深度理解）\_vue keepalived 配置详解-CSDN博客](https://blog.csdn.net/weixin_69422396/article/details/135479434?ops_request_misc=%7B%22request%5Fid%22%3A%22171532692116800213084865%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171532692116800213084865&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-3-135479434-null-null.142^v100^pc_search_result_base1&utm_term=Keep-alive%E5%8E%9F%E7%90%86&spm=1018.2226.3001.4187)

## OSI七层模型和TCP/IP四层网络模型

## 前端性能优化的手段

[前端性能优化9大策略（面试一网打尽）！-CSDN博客](https://blog.csdn.net/chaoPerson/article/details/130743570?ops_request_misc=%7B%22request%5Fid%22%3A%22171533222316800188598725%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171533222316800188598725&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-130743570-null-null.142^v100^pc_search_result_base1&utm_term=%E5%89%8D%E7%AB%AF%E6%80%A7%E8%83%BD%E4%BC%98%E5%8C%96%E7%9A%84%E6%89%8B%E6%AE%B5&spm=1018.2226.3001.4187)

## XMLHttpRequest

[XMLHttpRequest 状态码：readyState、status\_xmlhttp.readystate-CSDN博客](https://blog.csdn.net/qq_54954413/article/details/126185846?ops_request_misc=%7B%22request%5Fid%22%3A%22171533842316800225536182%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171533842316800225536182&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-3-126185846-null-null.142^v100^pc_search_result_base1&utm_term=XMLHttpRequest %E7%8A%B6%E6%80%81&spm=1018.2226.3001.4187)

## 浏览器渲染的原理

## 继承方式 手写代码

## 跨域：代理服务器的原理

[（最清楚）跨域问题使用代理服务器解决的原理\_代理服务器解决跨域问题原理-CSDN博客](https://blog.csdn.net/m0_74114657/article/details/134392111?ops_request_misc=%7B%22request%5Fid%22%3A%22171534582416800225575935%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171534582416800225575935&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-2-134392111-null-null.142^v100^control&utm_term=%E8%B7%A8%E5%9F%9F %E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%9A%84%E5%8E%9F%E7%90%86&spm=1018.2226.3001.4187)

## jsonp手写

## CORS要设置哪些属性

## Promise all race手写代码

## 响应式布局

## Axios原理

## Vue路由原理

[vue-router 路由超详细教程\_vue router-CSDN博客](https://blog.csdn.net/weixin_47124112/article/details/126730114?ops_request_misc=%7B%22request%5Fid%22%3A%22171540450916800182197029%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171540450916800182197029&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-126730114-null-null.142^v100^pc_search_result_base1&utm_term=Vue%E8%B7%AF%E7%94%B1&spm=1018.2226.3001.4187)

[hash模式与history模式-CSDN博客](https://blog.csdn.net/qq_37086980/article/details/138399837?ops_request_misc=%7B%22request%5Fid%22%3A%22171540533016800182717519%22%2C%22scm%22%3A%2220140713.130102334.pc%5Fall.%22%7D&request_id=171540533016800182717519&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~timeliness~default-6-138399837-null-null.142^v100^pc_search_result_base1&utm_term=hash%E4%B8%8Ehistory%E6%A8%A1%E5%BC%8F%E7%9A%84%E5%8C%BA%E5%88%AB&spm=1018.2226.3001.4187)

[vue-router 原理\_vue-router原理-CSDN博客](https://blog.csdn.net/AIROU_ao/article/details/128134863?ops_request_misc=%7B%22request%5Fid%22%3A%22171541718716800178521393%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171541718716800178521393&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-128134863-null-null.142^v100^pc_search_result_base1&utm_term=vue router%E5%8E%9F%E7%90%86&spm=1018.2226.3001.4187)

## 浏览器安全相关

xss攻击

csrf攻击

## Webpack原理，vite

## 浏览器存储 cookie、session、local、indexDB

[浏览器的四种本地存储方式-cookie,localStorage,sessionStorage,IndexedDB\_浏览器存储-CSDN博客](https://blog.csdn.net/qq_44947815/article/details/125341377)

## http状态码



## 浏览器缓存 状态码304

[浏览器缓存策略之强缓存与协商缓存\_强缓存浏览器服务器握手-CSDN博客](https://blog.csdn.net/DZY_12/article/details/105378730?ops_request_misc=%7B%22request%5Fid%22%3A%22171583885216800186523556%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=171583885216800186523556&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-105378730-null-null.142^v100^pc_search_result_base6&utm_term=%E6%B5%8F%E8%A7%88%E5%99%A8%E7%BC%93%E5%AD%98%E7%AD%96%E7%95%A5&spm=1018.2226.3001.4187)

[浏览器缓存（强缓存、协商缓存）及解决发版浏览器缓存问题\_vue项目如何解决浏览器强制缓存-CSDN博客](https://blog.csdn.net/weixin_44043530/article/details/130986530?ops_request_misc=&request_id=&biz_id=102&utm_term=%E6%B5%8F%E8%A7%88%E5%99%A8%E7%BC%93%E5%AD%98%E7%AD%96%E7%95%A5&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-130986530.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## Get post put delete等的区别

[Get、Post、Put、Delete和Patch的区别(简要概括)\_get post put delete patch-CSDN博客](https://blog.csdn.net/m0_59087223/article/details/132084367?ops_request_misc=&request_id=&biz_id=102&utm_term=Get post put%E7%AD%89%E7%9A%84%E5%8C%BA%E5%88%AB&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-3-132084367.142^v100^pc_search_result_base6&spm=1018.2226.3001.4187)

## css选择器优先级

# 项目

## Element-plus用到的组件记录

1. 修改密码时的侧边栏 el-drawer
2. 消息对话框 ElMessageBox
3. 消息提示框 ElNotification
4. 下拉菜单 el-dropdown
5. 各种图标 el-icon
6. 菜单栏 el-menu
7. 标签栏el-tabs
8. 布局Layout
9. 表格el-table
10. 悬浮时有提示的图标按钮el-tooltip
11. 开关 el-switch
12. Cascader 级联选择器，管理菜单权限列表时，需要指定权限或菜单的上级
13. Select 选择器，展示图标列表
14. Tree 树形控件，配置角色权限时展示权限列表
15. Popconfirm 气泡确认框，删除按钮被点击的时候弹除确认框
16. el-date-picker ，优惠券页面选择优惠券的开始和结束时间

17.用el-container容器进行布局

## 项目介绍

项目使用vue3进行开发，并用到了element-plus组件库和windicss样式库。用pinia存储用户数据，用axios与后端通信。实现的功能主要有登录退出；对商品，商品分类，商品规格的增加删除修改，对普通用户的管理，设置用户的会员等级；对管理员用户的管理，设置管理员的权限和角色,不同角色可以设置不同的权限，权限不同在系统里能看到的内容就不同；对订单的管理，修改订单评论

## 项目难点

项目本身可能并没有什么技术上很难实现的点，只是对于我个人而言，做项目的过程中是存在一些困难的，这种困难主要来自于对于框架不够熟悉。

第一点是对element-ui不熟悉，实现某个功能时不知道用哪个组件合适

第二点是对vue框架的使用不够熟悉。比如对于routerview、keep-alive和compent三个组件结合起来用，routerview包裹一个keep-alive，keep-alive包裹component，以这样的方式去实现组件缓存，开始时不太理解为什么要这么嵌套。

还有就是对于vue组合式api的使用不熟悉，不能很好对不同页面的相似功能抽象出公共的接口来提升代码复用率。

## 学到了什么

组合式api的使用、组件通信，v-model绑定到组件上、keep-alive

## 自我介绍

您好，很高兴参加今天的面试。我是中国科学技术大学的在读研究生，专业是软件工程。我本科毕业于江苏科技大学，专业是计算机科学与技术。我现在在中国科学院上海微系统与信息技术研究所实习，工作内容是利用python和pyqt开发桌面应用程序。由于现在后端岗位竞争比较激烈，在就业这方面我更看好要前端，于是开始了前端的学习。我在微系统所实习的过程中一直利用空闲时间学习前端，现在这里的实习马上技术了，我想找一个前端方向的实习，通过工作中的实践更深入地学习前端。虽然目前并未从事过前端开发的工作，但是我对自己的学习能力有足够的信心，我相信自己能快速掌握工作中要用到的知识和技能。并且我有一个很大的优势，我可以实习的时间长达一年，相比于短期实习生，我可以更好地融入团队，在项目中承担更多的责任。如果贵公司能给我一个机会，我一定会全力以赴，不断学习和进步，为团队的成功贡献自己的力量。

## 怎么学习前端的

看视频课，把视频里教的东西都自己敲一遍。遇到困惑就百度或者问GPT。另外牛客上有前端的练习题目，涉及到html、css、js，这些题目我都做了，对学习也是有帮助的。

# 八、已投递

## 1.已投递

1.滴滴

[滴滴 - 校园招聘 (didiglobal.com)](https://campus.didiglobal.com/campus_apply/didiglobal/96064" \l "/candidateHome/applications)

2.字节

[应聘记录 (bytedance.com)](https://jobs.bytedance.com/referral/pc/position/application?token=MTsxNzE0Mjc4MzAwMjA2OzczMTk2ODY0NDkzNTA4NTQxNzE7NzMyMDUwMDUzMjgxMTU0ODk3ODsx)

3.腾讯

[应聘进度 | 腾讯校招 (qq.com)](https://join.qq.com/progress.html)

4.美团

[个人中心 | 美团招聘 (meituan.com)](https://zhaopin.meituan.com/web/personalCenter/deliveryRecord)

5.京东

[京东校招 (jd.com)](https://campus.jd.com/home" \l "/myDeliver?type=present)

6.快手

[快手校招 - 快手校招-投递记录 (kuaishou.cn)](https://campus.kuaishou.cn/recruit/campus/e/" \l "/campus/my-apply)

7.百度

[百度校园招聘 (baidu.com)](https://talent.baidu.com/jobs/center)

8.飞猪

[飞猪招聘官网 (fliggy.com)](https://career.fliggy.com/personal/campus-application?lang=zh)

9.小红书

[投递记录 (xiaohongshu.com)](https://job.xiaohongshu.com/record/campus)

10.合合信息

[合合信息招聘门户 (zhiye.com)](https://intsig.zhiye.com/login?goto=personal/deliveryRecord)

11.momenta

[应聘记录 - 加入Momenta (feishu.cn)](https://momenta.jobs.feishu.cn/intern/position/application)

12.钉钉

[钉钉招聘官网 (dingtalk.com)](https://talent.dingtalk.com/personal/campus-application?lang=zh)

13.蚂蚁

[蚂蚁集团招聘官网 (antgroup.com)](https://talent.antgroup.com/personal/campus-application)

14.东方财富

[东方财富-校园招聘 (mokahr.com)](https://app.mokahr.com/campus-recruitment/eastmoney/57971" \l "/candidateHome/applications)

15.阿里国际数字商业 广州

[阿里国际数字商业集团招聘官网 (alibaba.com)](https://aidc-jobs.alibaba.com/personal/campus-application?lang=zh)

16.哔哩哔哩

## 2.测评

钉钉

蚂蚁

京东 [人才评估 (ceping.com)](https://360buy.ceping.com/pc?elink=zGiefBc0rGbL/Ak4w/CqdspRa5fYRRJpUG0RUedSHKPZ3zBhlk5hhUxZa/bpy9ViXhPJbWdwhVfvJ5V78Yq3zg==" \l "/promise)

## 3.笔试

七牛云 5.8 19:30 [2024年春招前端笔试\_牛客 (nowcoder.com)](https://exam.nowcoder.com/cts/17342254/summary?id=2A3086DF63FDA6E95D2B1AAD67C4DCC3)

## 4.面试

美团 简历挂

快手 简历挂

阿里国际数字商业 简历评估通过待发面试链接

飞猪 一面5.10 10:00 一面挂

字节 一面5.10 19:00 一面挂

钉钉 一面5.11 10:00 一面挂

蚂蚁 一面5.11 14:00 一面后没动静

腾讯 一面5.11 19:00 一面挂

滴滴 一面5.15 18:00 二面5.15 19:13 二面挂

哔站 一面5.16 19:00

魔门塔 一面5.17 14:00