**2017年秋招总结**

目录

[一、简历 3](#_Toc494300359)

[1.基本信息 3](#_Toc494300360)

[2.教育背景 4](#_Toc494300361)

[3.专业技能 4](#_Toc494300362)

[4.项目经验 4](#_Toc494300363)

[5.校内实践 4](#_Toc494300364)

[6.工作/实习经历 4](#_Toc494300365)

[7.获得奖项 4](#_Toc494300366)

[8.技能证书 4](#_Toc494300367)

[9.自我评价 5](#_Toc494300368)

[10.简历模板 5](#_Toc494300369)

[二、内推 5](#_Toc494300370)

[三、笔试 5](#_Toc494300371)

[1.学习路线 5](#_Toc494300372)

[2.笔试经验 6](#_Toc494300373)

[四、面试 6](#_Toc494300374)

[1.面试注意事项 6](#_Toc494300375)

[2.搜狐内推电话面试（前端工程师） 6](#_Toc494300376)

[3.华为大牛场面试（IT应用软件开发工程师） 7](#_Toc494300377)

[4.好未来视频面试（PHP后端开发工程师） 7](#_Toc494300378)

[5. ThoughtWorks代码作业（软件开发工程师JS方向） 8](#_Toc494300379)

[6.大疆电话面试（web前端工程师） 8](#_Toc494300380)

[7搜狐霸面（web前端开发工程师） 8](#_Toc494300381)

[8.百度现场面试（web前端开发工程师） 9](#_Toc494300382)

[9.五八集团现场面试（web前端开发工程师） 9](#_Toc494300383)

[10.用友现场面试（web前端开发工程师） 10](#_Toc494300384)

[11.去哪儿网现场面试（web前端开发工程师） 10](#_Toc494300385)

[12.腾讯现场面试（web前端开发工程师） 11](#_Toc494300386)

[13.美团点评现场面试（web前端开发工程师） 11](#_Toc494300387)

[14.附上秋招连跪表 13](#_Toc494300388)

[五、前端技术路线 13](#_Toc494300389)

[1.计算机基础 13](#_Toc494300390)

[2.语言基础 13](#_Toc494300391)

[3.专业基础 14](#_Toc494300392)

[4.框架源码 14](#_Toc494300393)

[六、笔试刷题知识点 14](#_Toc494300394)

[1.链表 14](#_Toc494300395)

[2.栈 15](#_Toc494300396)

[3.队列 16](#_Toc494300397)

[4.数组 16](#_Toc494300398)

[5.字符串 17](#_Toc494300399)

[6.树 18](#_Toc494300400)

[7.堆 18](#_Toc494300401)

[8.散列（哈希表） 19](#_Toc494300402)

[9.图 19](#_Toc494300403)

[10.算法 20](#_Toc494300404)

[11.插入排序 20](#_Toc494300405)

[12.希尔排序 20](#_Toc494300406)

[13.堆排序 20](#_Toc494300407)

[14.归并排序 20](#_Toc494300408)

[15.快速排序 20](#_Toc494300409)

[16.递归 21](#_Toc494300410)

[17.浏览器 21](#_Toc494300411)

[18.H5 22](#_Toc494300412)

[19.前端 23](#_Toc494300413)

[20.网络 24](#_Toc494300414)

[21.数据库 24](#_Toc494300415)

[七、前端面试题汇总- Html+H5 24](#_Toc494300416)

[1. cookie/localStorage/sessionStorage/userData/session 24](#_Toc494300417)

[2.DOM结点增删查改 25](#_Toc494300418)

[八、前端面试题汇总-CSS+CSS3 26](#_Toc494300419)

[1.垂直居中方法 26](#_Toc494300420)

[2.flex弹性布局 27](#_Toc494300421)

[3.CSS实现图片轮播 27](#_Toc494300422)

[4.盒子模型 29](#_Toc494300423)

[5.display:none;和visibility:hidden和overflow:hidden区别 29](#_Toc494300424)

[6.BFC 29](#_Toc494300425)

[7.overflow属性 30](#_Toc494300426)

[8.边框重叠 30](#_Toc494300427)

[9.三列布局 30](#_Toc494300428)

[九、前端面试题汇总-JS+ES6 30](#_Toc494300429)

[1.JS返回类型的方法 30](#_Toc494300430)

[2.instanceof原理 31](#_Toc494300431)

[3.js设计模式 31](#_Toc494300432)

[4. JS事件onchange、oninput、onpropertychange区别 31](#_Toc494300433)

[5.this指向 31](#_Toc494300434)

[6.bind实现原理 31](#_Toc494300435)

[7.new一个实例，中间经过哪几个过程 32](#_Toc494300436)

[8.深/浅拷贝 32](#_Toc494300437)

[9.原型链 32](#_Toc494300438)

[10.闭包 33](#_Toc494300439)

[11.作用域链 33](#_Toc494300440)

[12.Ajax原理 33](#_Toc494300441)

[13.js内存泄漏的操作 33](#_Toc494300442)

[14.setTimeout和setInterval原理 34](#_Toc494300443)

[15.js改变css样式 34](#_Toc494300444)

[16.js继承 35](#_Toc494300445)

[18.js设计模式 36](#_Toc494300446)

[十、前端面试题汇总-专业基础知识 36](#_Toc494300447)

[1.域名从输入到响应过程 36](#_Toc494300448)

[2.进程和线程区别 37](#_Toc494300449)

[3.mvvm数据双向绑定原理 37](#_Toc494300450)

[4.前端优化规则 38](#_Toc494300451)

[5.内联图片 38](#_Toc494300452)

[6.浏览器如何完成渲染 38](#_Toc494300453)

[7.前后端分离 38](#_Toc494300454)

[8.跨域的方式有几种 39](#_Toc494300455)

[9.JSONP原理 39](#_Toc494300456)

[10.webpack脚手架 39](#_Toc494300457)

[十一、前端面试题汇总-框架 40](#_Toc494300458)

[1.BootStracp栅格布局 40](#_Toc494300459)

[2.Vue生命周期&钩子函数 40](#_Toc494300460)

[2.Vue组件通信 40](#_Toc494300461)

[3.Vue双向数据绑定 40](#_Toc494300462)

[4.项目 41](#_Toc494300463)

[十二、前端面试题汇总-网络 41](#_Toc494300464)

[1.Socket通信 41](#_Toc494300465)

[2.OSI七层模型和TCP/IP五层模型 42](#_Toc494300466)

[3.TCP/IP协议 42](#_Toc494300467)

[4.Http状态码 43](#_Toc494300468)

[5.TCP和UDP区别 44](#_Toc494300469)

[6.Http头信息 44](#_Toc494300470)

[7.https加密过程 44](#_Toc494300471)

# 一、简历

## 1.基本信息

姓名、求职目标（只用写一个）、年龄、性别、毕业院校、专业、手机号、邮箱

Github优秀的可以给链接

个人博客优秀的可以给链接

个人作品优秀的可以给链接或者附件

## 2.教育背景

由近到远依次列出

时间、学校、专业、学位

## 3.专业技能

语言、框架、操作系统、数据库、计算机网络、开发工具

技能层次：了解、熟悉、熟练、精通（精通别写，切记，除非你可以碾压面试官）

## 4.项目经验

由近到远依次列出

时间、项目名称、负责工作

项目描述（简练，突出重点）

责任描述/项目成果（3-5条，以列表的形式，主次清晰，突出重点）

## 5.校内实践

能写则写，大小都可以，加分

## 6.工作/实习经历

最好写与求职相关的工作经历，加分

不相关但可以体现能力的也可以写，可能加分

（写重点）

## 7.获得奖项

有多少写多少

（幼儿园小红花什么的就不用写了）

## 8.技能证书

有多少写多少

CET4/CET6分数高的可以具体写出来

## 9.自我评价

不要抄网上模板（减分）

写你个人的亮点，你的发展规划，你的兴趣爱好

## 10.简历模板

五百丁

# 二、内推

在7月份到9月份关注内推信息，几乎每家公司都有大量内推机会

内推招人多，很多免笔试，比校招容易，进大厂机会大，当然，需要提前准备

网站：

牛客网/讨论区

清水河畔/招聘区

微信公众号：

每日校招

# 三、笔试

## 1.学习路线

基础：

数据结构

算法

语言基础

数据库

计算机网络

操作系统

Web前端：

JS/CSS/Html基础

ES6

H5

CSS3

jQuery

NodeJs

Vue/React框架

书籍：

剑指offer

数据结构与算法

网站：

牛客网专项练习

牛客网真题练习

牛客网编程练习

LeetCode

W3C官网

## 2.笔试经验

笔试最好在牛客网，其次赛码网，其他OJ都比较糟糕

编程题先用数学方法解决，暴力解法最后考虑

# 四、面试

## 1.面试注意事项

（1）技术岗面试短袖、长裤、布鞋即可

（2）准时

（3）休息好，吃好，面试前吃一个口香糖

（4）面试时保持头脑冷静，语言清晰，逻辑清楚

（5）成功失败都别在面试过程中表现出情绪

（6）不轻易说不知道，冷静思考，即使做不出来也要说出思路，分析难点

（7）全程礼貌

（8）一面一般是基础面试，问基础，根据笔试题目考察真实度

（9）二面一般是技术主管面试，问项目，问基础

（10）三面一般是总监面，问项目，问架构，问源码

（11）四面hr面，考察性格，布置坑点，问基本情况，衡量薪资待遇

## 2.搜狐内推电话面试（前端工程师）

这是我最早面试的公司，网上内推，八月初面试

约的是下午四点电话面试，很是准时，四点刚到手机就响了，我有点紧张

面试官听声音是一个比较年轻的技术人员，刚开始闲聊了几句

他说看我的简历比较适合做后端，只问我JS的问题（他所在的部门应该主要做JS）

判断数组类型方法，跨域，JSONP原理，this指向，bind实现，new一个构造器中间的过程，深拷贝、浅拷贝，webpack，描述原型链...

问题都很基础，可惜我回答得很糟糕，面试官人很好，一次又一次引导我

总结：基础非常重要，早做准备

## 3.华为大牛场面试（IT应用软件开发工程师）

我是以炮灰的心态去面华为的，华为现场面试安排得井井有条，有小吃，有早餐

现场还有一个HR负责缓解我们紧张的情绪，简单解答疑问，每人送了一本书

面试推迟了半小时

一面是专业面，面试官很年轻，看起来不到30岁，很是客气

面试官语气很温和，开始在看简历，看我没事做就让我手写字符串拷贝，用C

之后是一系列的手撸代码：PHP写冒泡排序，快速排序，JS写正则匹配题...

然后问项目，从第一个开始，CNCUI项目的核心，项目结构，原理

第二个，网站亮点，原理，如果这样改动会怎么样，为什么，能不能那样改...

第三个，为什么重构网站，优点在哪，怎么优化的，遇到什么难点...

问实践，实验室管理主要做什么，用什么软件...

一面40分钟，聊得很好，通过

二面是综合面，面试官比较年长，估计是部门Boss，很客气

面试官满面笑容，问为什么年龄这么小，家里情况，学习成绩，工作意向

你们学校的人最想去哪家公司？（必须说华为！）

给你华为网易百度等等offer你会去哪家？（华为！）

面试官很开心，我很镇定

华为给你offer你会来吗？（会来）一定吗？（肯定）

面试官很开心，我很镇定

接下来问了平时学习的时间安排，工作习惯，对加班的看法，对云端的一些理解，SAAS，有没有用过云服务器（我报的部门是云核心网）...

最后面试官站起来握手，又问了一句：如果华为给你offer你一定会来吗？（一定），记住你说的话，我选择你，你也要选择我（天真的我以为稳了，开始了一个多月的资源池排序...）

总结：运气很重要，所以没过不一定都是实力不够，坑不能踩，保持镇定，思路清晰

## 4.好未来视频面试（PHP后端开发工程师）

面试在华为面试同一天的下午，由于头天晚上只睡了2个小时，到下午精神状态已经很糟糕，思维僵硬（所以面试前一定要休息好，切记切记！）

面试很准时，面试官是一个中年男子，满面笑容，但是我一开始就感觉他对我不是很感兴趣

视频面试是在牛客网的平台上进行的，面试官可以在平台上出题，我的每一步代码他都能看到

开始是基础问题，我常用的PHP数组和字符串函数有哪些（写了几十个），http状态码，握手挥手...

手撸代码：

1.递归的方法判断字符串是否对称

2.遍历文件夹中的文件和文件夹，如果有文件夹，继续遍历

3.图片操作

4.数组去重

http请求类型有哪些，每种请求的区别，数据库操作

没问项目，基础答得不错，手撸代码都很基础，但是写得不好，思维僵硬了

总结：面试前要休息好，基础要扎实

## 5. ThoughtWorks代码作业（软件开发工程师JS方向）

ThoughtWorks代码作业有难度，制作一个羽毛球馆收费系统（具体我会放在外面）

我从下午2点写到凌晨1点，完成了代码作业

第2天早上起床就开始进行测试，修改，完善，优化，一直到中午才完成提交

总共300行代码，注释详细，操作文档也写了

然而还是没给我面试机会，这让我很是失落，尽力了，完成了

总结：动手去做，不问前程，英语要学好（大概是我6级没过，所以没有给面试机会）

## 6.大疆电话面试（web前端工程师）

电话面晚了20分钟到来，面试官听声音是一个很年轻的小哥，先跟我道了歉，我表示没关系

能感觉到面试官很紧张，说话会口吃，会抖，手动微笑~

自我介绍

三列布局，左右固定，中间自适应

实现一个函数，奇数次调用返回1，偶数次调用返回2

垂直水平居中方法有哪些

JS类型判断，position属性

纯CSS实现图片轮播

动画animation

快排原理，希尔排序原理

vue生命周期，父子通信

移动端布局，H5语义化理解

深拷贝、浅拷贝

你有什么想问我的（您觉得我哪方面需要加强）...

答得不好，面试官很宽容，安慰我继续努力，他说他一紧张就口吃...

总结：不仅要基础扎实，具体的要求也要能灵活实现

## 7搜狐霸面（web前端开发工程师）

搜狐创新研究院，总监面，面试官气场强大，同时面我和另一个后端的同学

带了笔没（我没有，另一个带了）

后端的先手写一道算法题（字符串匹配，不用正则，我想到的是KMP算法，不过那位同学直到我面试结束也在埋头苦写...）

前端提问了

跨域，缓存，前端优化，内联图片，图片地图，https和http区别，https加密过程，堆排序过程，position属性，webpack，垂直居中，flex布局，H5语义化，ES6，promise原理...

面试官说可以，后面还会有电话面一次技术基础，hr说应该问题不大（天真的我就信了，然后等了很久很久...）

总结：有机会就要去争取，准备要充分，知识点要深入理解

## 8.百度现场面试（web前端开发工程师）

坐地铁去市区一个酒店面试，同行的还有一个妹子，她投的是测开

面试很准时，工作人员通知去XXX房间面试

面试官是一个很漂亮，气场强大的女人

我去面试的时候里面还有一个正在面的，我站在门口轻轻敲了敲门，面试官示意我稍等

面试官气质偏冷，带着微笑，又给人距离感

面试官从CSS、HTML、JS到CSS3、HTML5、ES6问得很细节

CSS选择器，权重，优先级，伪元素，垂直居中，position属性及用法原理，display...

CSS3动画，flex布局，3D，多列布局...

HTML表单元素，input disable，手写Ajax过程，自适应布局...

HTML5动画，新增类型...

JS函数，类型判断，forEach，reduce，this绑定，手写对象深拷贝，手写数组翻转...

ES6数组解构，字符串模板，手写promise封装异步Ajax，let、const，class，箭头函数...

你有什么想问我的（您觉得我哪方面需要加强）...

面了80分钟，面试官说我笔试成绩挺高的，问得不难，但是很细，把以前没注意的重点都翻了一遍，挂得我很服气，这次面试收获很大

总结：基础很重要，不是说说而已，要将每个要点理解透彻，ES6要重点看

## 9.五八集团现场面试（web前端开发工程师）

在58宣讲会拿到了PASS卡，免笔试

面试在市区一个酒店，等了一个多小时

面试官是一个30左右的小哥，看起来兴趣不是很高

自我介绍，怎么没有笔试，怎么拿到的PASS卡

浏览器渲染原理，什么情况下产生reflow和repaint，他们的区别

前端优化，缓存有几种，原理，如果缓存要改变怎么做

跨域的方式，各自的优劣，手写JSONP跨域，web workers能不能跨域

webpack脚手架用过哪些，有没有用过XXX，脚手架用途

http请求类型，请求过程，状态码

XXS攻击原理，怎么防御，SQL注入

H5动画，SVG和Canvas

设计模式，工厂模式和单例模式，观察者模式

ES6知道哪些，手写promise，箭头函数，class继承，高阶函数

Vue生命周期，组件通信，双向数据绑定，用到哪些设计模式

你有什么想问我的（您觉得我哪方面需要加强）

...

问得比较全面，答得还可以，最后跟我说ES6不够熟练，ES6熟练的话公司可以直接拿来干活，不然要培养2周，把我挂了，这是我最不服气的一次面试，不过也有很多收获

总结：运气很重要，所以挂了也不用灰心，从面试中知道你的弱点，回去后马上加强学习，ES6真的重要（以后会要求更高），知识点不仅要知道，还要深入理解、扩展

## 10.用友现场面试（web前端开发工程师）

宣讲会结束后现场笔试，选择题多选，选错不得分，5个填空题，3个编程题

选择题答得一般，填空题不错，编程题对2个，还有1个前端布局题做完之后才想到

笔试结束当天晚上收到面试通知，从工作人员处得知笔试成绩不错

一面hr面

自我介绍

问了家庭，工作意向，对加班的看法

目前有没有offer（没有，华为在资源池），正在面哪些（百度、搜狐、美团、京东）

如果百度和我们都给你offer，你怎么选择（这个是我自己挖的坑，当时冷汗下来了，1秒钟之后，我从两家公司的核心业务，对前端的重视程度，前端上升空间分析了一波，选择用友）

hr很满意，说我对问题看得很透彻（- -！汗）

你最不喜欢什么样的人（情商低的）

不喜欢直接的？（直接的没问题，我指的是做事不考虑别人感受的）

hr笑了，你是不是被人伤过（一笑带过）

你和别人对比，你的优势，你的缺点

你有什么想问我的（您对我有什么建议）

...

二面总监面

自我介绍

项目，架构，画出来，为什么选择Laravel，为什么选择Vue，为什么不选择React，Vue特性

别人对你的看法，你的项目亮点，难点，你是怎么解决的，看过哪些源码

你平时学习方法，你实践里面有销售，你是怎么做的

数据库操作

手写算法...

百度地图怎么判断交通堵塞

你还有什么要问的（您觉得我哪方面需要加强）我觉得你很聪明，项目逻辑很清晰

晚上还会有一轮电话面问技术问题...

三面技术面

面试官是个年轻小哥，很客气，全程用您称呼...

问的都是基础，包括跨域，缓存，优化，ES6，前后端分离...

第二天通知签offer

总结：镇定，不卑不亢，不踩坑，不挖坑，基础要打牢

## 11.去哪儿网现场面试（web前端开发工程师）

在市区一个酒店面试，面试很准时

一面应该是个Boss，很礼貌

手写：

（1）实现bind函数，实现新的bind函数bind（fn，object）

（2）数组去重

（3）给promise添加一个方法，使：无论promise返回resolve还是reject，调用该方法都能输出结果，该方法只有一个回调函数

（4）实现网页版计算器，你会怎么布局，你会怎么实现，手写核心代码

（5）flex布局的原理，flex布局实现垂直水平居中，其他的flex属性，align-items和align-content区别

（6）将一个网页上所有元素标签以字符串形式存入数组中，不能重复

（7）数组逆序，不用reverse

问题：

（1）深拷贝/浅拷贝

（2）基础数据类型

（3）==和===各自用法

（4）[1]==[1]？为什么？如果是引用呢，为什么？

（5）状态码：301,302，会不会再次请求，304,401

（6）平时怎么学习

（7）看过的框架源码

（8）vue优点

（9）vue组件通信，父子通信，父调用子，兄弟组件通信

（10）双向数据绑定原理，ie/非ie

（11）脏值检测和数据劫持的优缺点

（12）项目难点，怎么解决的

（13）session和cookie区别，session怎么释放，服务端怎么识别客户端

（14）有没有用过react

（15）前端优化

（16）缓存、跨域

（17）this指向问题

（18）setTimeout机制

一面一个半小时，答得不错

二面

hr问了些家庭情况，工作意向，XX和我们同时给你offer，怎么选择，为什么，平时怎么学习，从项目中有什么收获，问了10分钟，现场签sp

## 12.腾讯现场面试（web前端开发工程师）

进去就感觉到面试官对我兴趣不大，面试官靠在沙发上，表情冷淡

自我介绍

为什么选择Vue+Laravel，Vue特性，生命周期，组件通信，双向数据绑定怎么实现

跨域的几种方式，各自的优缺点

缓存方式，缓存日期

手写JS继承（写了5种）

设计模式

http请求过程

手写Ajax请求原理

动画

PC端和移动端区别

nodejs，nodejs和php对比

setTimeout原理

webpack

ES6...

回答得不错，然而没给过

另一个全程聊人生的同学过了，而且二面他还是在聊人生- -！

总结：挂了不用灰心，继续学习，总能遇到伯乐

## 13.美团点评现场面试（web前端开发工程师）

面试很准时，提前10分钟到我了

一面

面试官是个中年男子，很和蔼，全程笑脸，没问基础

项目，项目架构，画结构（详细）

项目难点，怎么解决的，平时学习方法

改项目，怎么实现

移动端接口怎么实现

实现一个web端学习平台，有课程、作业、评价、得分等等功能

需要多长时间，你会怎么实现，画架构图，实现过程

...

直接让我去二面

二面

面试官不到30岁，表情严肃，偶尔有笑容，其实很宽容

JS基础，手写代码，在浏览器开发工具上码代码，现场编译看结果

错误的地方会耐心解释，指导

变量置换，不用新的变量，ES5的解法，ES6的解法，智力型解法

给出一个页面，怎么按顺序提取其中的元素，不重复，手写代码实现，现场编译

海量列表，点击改变对应元素样式（事件代理），手写代码

Vue事件机制，组件通信，数据双向绑定ie、非ie，数据劫持，脏值检测

...

三面，总监面

面试官是Boss，亲和中带着冷漠

对着简历从头到尾提问

Vue要点，生命周期，钩子函数，axios（promise+ajax封装）

BootStrap栅格布局，自适应布局

画MVC架构和MVVM架构图，工作过程，各自优点

Laravel优点，+Vue的好处

面向对象，三大特性

项目，难点，亮点

销售经历，怎么计划，谁组织的，有没有因为钱产生矛盾，为什么不继续做下去

为什么选择前端

...

四面，hr面，压力面

hr看似没看过我的简历，乱说一通，全程否定（我保持冷静）

自我介绍

你是做C还是C++的（冷静地告诉他，我做前端开发）

前端不如后端，你为什么选择前端（冷静，跟他分析一波后端确实不错，前端也很重要blabla）

平时怎么学习，项目难点怎么解决

家里情况，工作地点意向

手上有没有offer，哪些，给的待遇多少，有哪些在面试

如果XXX和我们都给你offer你会怎么选择，为什么

XX不行，XXX已经开始没落了，你确定要去那里？我觉得你可以选择XXXXXXX...

（冷静，微笑，听他怎么说）

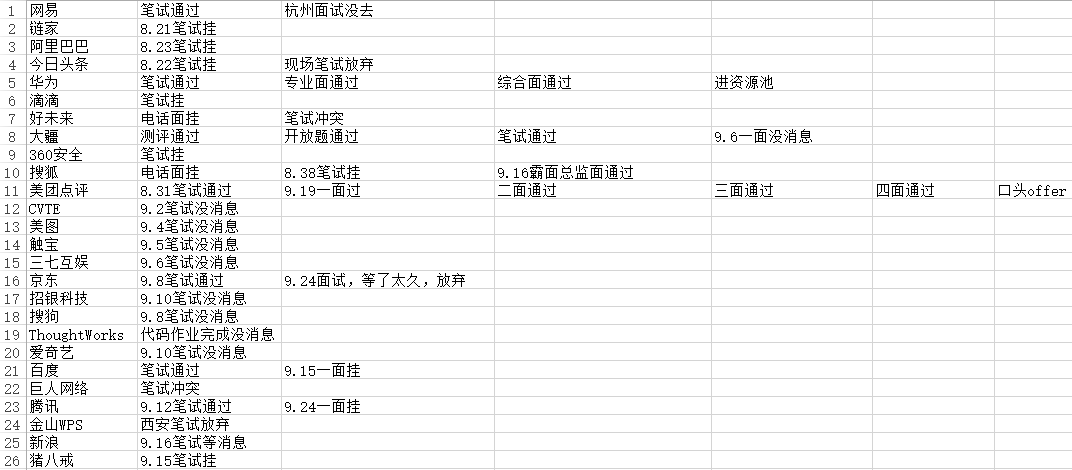
你觉得三年时间内一个技术管理能有多少薪资，你不考虑转技术管理吗（这是坑，别踩，坚决表示只做技术，对管理岗位没有非分之想）

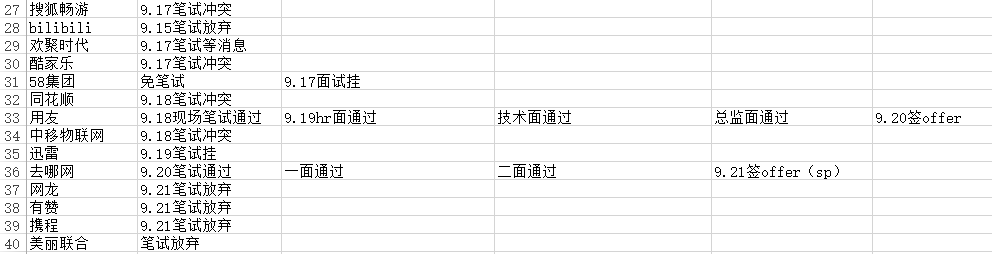
还是要感谢你来参加美团的面试...（- -！what？）（冷静，跟他说谢谢）

第二天就发来了口头offer...

总结：没有写出来的代码题别灰心，说出你的思路以及难点在哪里，面试官要看得还有你的思维和面对问题的态度，永不放弃，面对压力不要焦躁，冷静以对，看他怎么说

14.附上秋招连跪表





# 五、前端技术路线

## 1.计算机基础

数据结构：数组、字符串、链表、栈、队列、堆、哈希表、树、图

算法：快排、冒泡、插入、希尔、选择、堆排序、基数排序、桶排序、KMP、二分查找

数据库：增删查改

操作系统：linux基本操作

网络：http、https、握手挥手、网络分层、每一层的协议及用途

学习途径：看数据结构和算法，看剑指offer，牛客网刷题，LeetCode刷题

## 2.语言基础

JS、ES6、Html、H5、CSS、CSS3

学习途径：W3C官网、慕课、百度云课堂、网易云课堂、我的笔记

ES6：<http://es6.ruanyifeng.com/>

## 3.专业基础

jQuery（JS库）、NodeJs（前后端架构）、npm（开发工具）、webpack（前端资源整合工具）、Git（开源项目管理）

学习途径：W3C官网、慕课、百度云课堂、网易云课堂、CSDN、博客园、简书、stack overflow

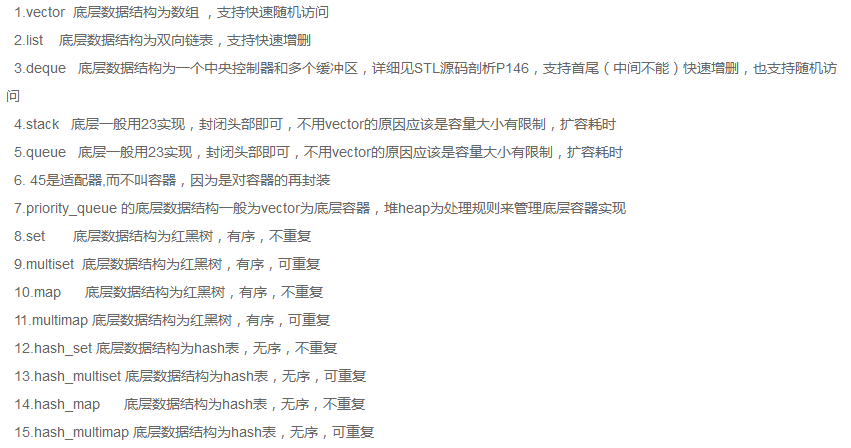
## 4.框架源码

Vue、React、Angular、BootStracp

学习途径：Vue、React、BootStracp官网及中文社区、以上视频网站

# 六、笔试刷题知识点

## 1.链表



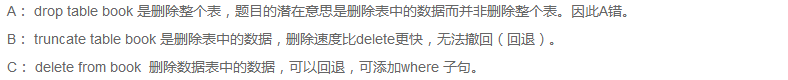


▲在（非空）广义表中：

1、表头head可以是原子或者一个表

2、表尾tail一定是一个表

## 2.栈



▲深度用栈，广度遍历用队列

▲n圆盘的Hanoi塔移动次数f(n) = 2^n – 1

▲以下选项中应用到栈：递归、快速排序（非递归程序用栈实现）、表达式求值、树的遍历

▲通常使用栈来处理函数或过程的调用

▲堆栈溢出一般是由什么原因导致的？循环的递归调用，大数据结构的局部变量

▲关于堆和栈的区别：

栈的大小是固定的，堆的大小受限于系统中有效的虚拟内存

栈的空间由系统决定何时释放，堆需要自己决定何时去释放

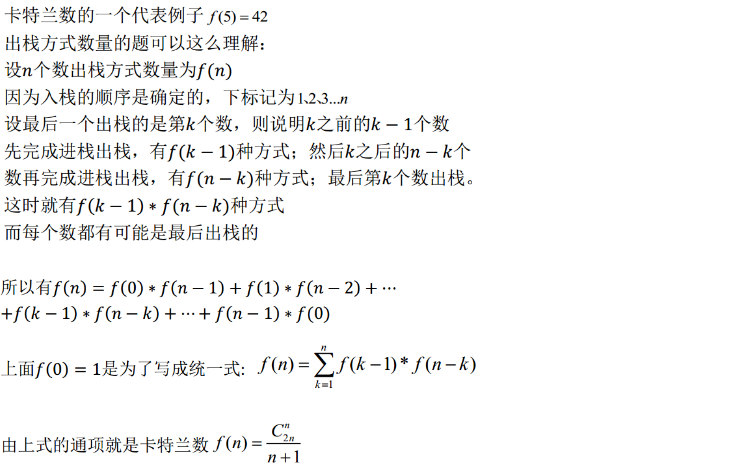
堆的使用容易产生碎片，但是用起来最方便

▲将一个递归算法改为对应的非递归算法时，通常需要使用栈

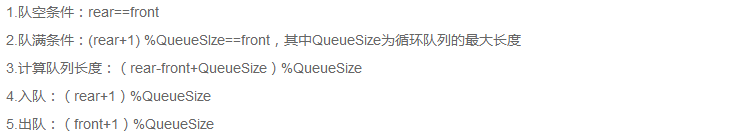
▲尾递归可以转换为循环的形式消除栈

▲DFS是图的深度优先遍历算法，用栈

▲n个元素出栈顺序



## 3.队列

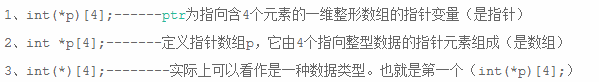


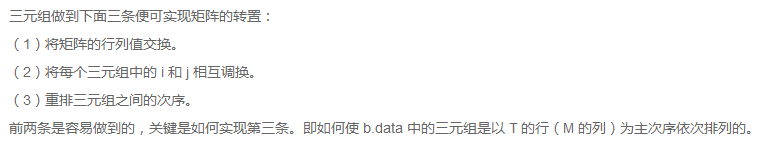
▲在链队列中，即使不设置尾指针也能进行入队操作

▲开源软件中经常被用作队列的是：Redis

▲有n个元素的序列，若使用筛选法建堆，则从位置为n/2取下整的元素开始建堆

## 4.数组





▲稀疏矩阵压缩的存储方法是: 三元组、十字链表

三元组：稀疏矩阵压缩方式，将矩阵中非0元素的行列号以及值构成一个三元组（行号，列号，非零值）稀疏矩阵转置，三元组行列号互换，并且按照原列号重新排序

十字链表：除了有行列号、值以外还有指向同一行和列下一个非零结点的指针域

▲对矩阵压缩存储是为了减少存储空间

▲静态查找表的查找方法主要有：有序表查找及有序表、静态树表、索引顺序表等查找方法

▲数组特殊矩阵压缩存储

（1）对称矩阵

在n阶对称矩阵中，Aij=Aji

我们只用为每一对对称元素分配一个存储空间

所以可以将n2个元素压缩存储到n(n+1)/2个元的空间中

一维数组A[n(n+1)/2]作为n阶对称矩阵的压缩存储结构

a[k]与矩阵元素之间存在着对应的关系：当i>=j时，k=i\*(i+1)/2+j;当i<j时，k=j\*(j+1)+i;

（2）上下三角矩阵

上、下三件矩阵是指矩阵上（下）三角（不包括对角线）中的元素均为常数c或0的n阶矩阵。则除了和对称矩阵一样，只储存其下（上）三角中的元素之外，再加一个储存常数c的储存空间

（3）对角矩阵

对角矩阵是指所有的非0元素都集中在以主对角线为中心的袋装区域中，即除了主对角线上和直接在对角线上、下方若干条对角线上的元素之外，所有其他的元素皆为0

假设对角矩阵的半宽为d,则对于n阶矩阵需要n\*(2\*d+1)-d\*d-d+1个储存空间

## 5.字符串

▲KMP算法



对应的部分匹配值指最后一位能匹配值所对应的部分匹配值！

▲字符串最后以 ’ \0 ’ 结束。strcpy将'1234\0'复制到目的地址

strlen()遇到 ’ \0 ’ 停止

## 6.树

▲n个结点的完全二叉树深度为[logn]+1

▲深度为K的满二叉树叶子结点数为2^(K-1)，前K-1层结点称为分支节点

▲最小生成树可以有多个， 他们的权值相等且最小即可

▲二叉树与树是不同的，二叉树不等价于分支树最多为二的有序树。当一个结点只包含一个子节点时，对于有序树并无左右孩子之分，而对于二叉树来说依然有左右孩子之分，所以二叉树与树是两种不同的结构

▲二叉树的度是当前节点下方边的总和，可能为：0、1、2

n个结点的二叉树，有n-1条边，度总和为n-1

假设1度结点为x个，2度为y，0度为z

x+2y=n-1,x+y+z=n

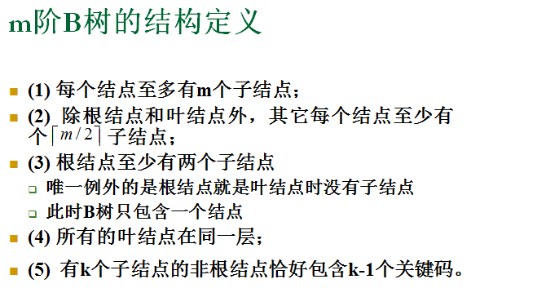
推出y=z-1

▲线索二叉树结点，左子树为空时，左指针指向后序遍历前驱

右子树为空时，右指针指向后序遍历后继

▲哈夫曼树，左右孩子权值之和为父结点权值，孩子权值只能小于父结点权值

▲n阶B-树，每个树根结点中所含的关键字数目最多允许为n-1个，最少允许为n/2-1个（向上取整ceil）



▲AVL二叉平衡树的插入，4种情况破坏平衡，单双旋转

## 7.堆

▲堆是完全二叉树，不保证是平衡二叉树，因为堆中左右子树的高度差并不保证小于等于1

▲优化队列是逻辑结构，堆是数据结构，数组是存储方式

▲插入新元素、删除堆顶、大根堆、小根堆

▲堆化数组时间复杂度为O(n)，堆排序时间复杂度为O(n\*logn)

## 8.散列（哈希表）



▲Hash 表的查找效率取决于散列函数、处理冲突的方法和装填因子。显然，冲突的产生概率与装填因子（表中记录数与表长之比）的大小成正比，即装填得越满越容易发生冲突

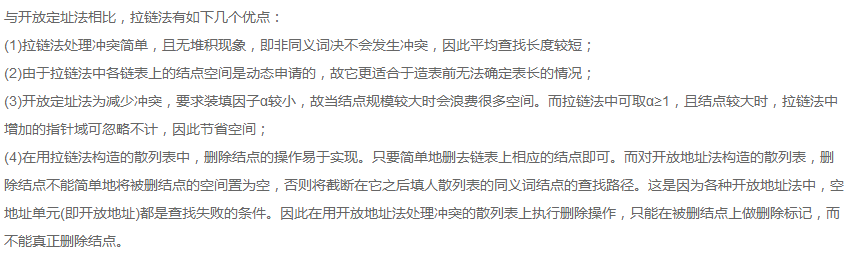
▲开哈希表-------链式地址法

闭哈希表-------开放地址法

▲处理哈希冲突方法有：开放地址法（线性探测法、线性补偿探测法、随机探测），拉链法，公共溢出区，再散列法

▲既希望较快的查找又便于线性表动态变化的查找方法是索引顺序查找

▲有B+Tree/Hash\_Map/STL Map三种数据结构。对于内存中数据，查找性能较好的数据结构是（Hash\_Map），对于磁盘中数据，查找性能较好的数据结构是（B+Tree）



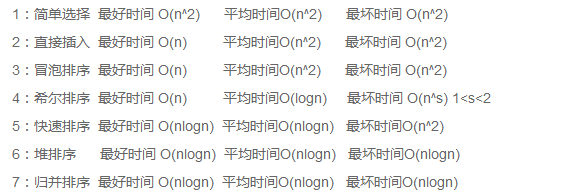
## 9.图

▲无向图存储：邻接矩阵、邻接表、多重邻接表

有向图存储：邻接矩阵、邻接表、十字链表

▲数据流图中带有箭头的线段表示的是数据流

## 10.算法



11.插入排序

插入排序由N-1趟排序组成

对于P=1趟到P=N-1趟，插入排序保证从位置0到位置P上的元素为已排序状态

遍历元素，把每个元素左移到当前它应该出现的位置

12.希尔排序

它通过比较相距一定间隔的元素来工作

各趟比较所用的距离随着算法的进行而减少

直到比较相邻元素的最后一趟排序为止

（缩小增量排序）

增量序列：h1，h2…

13.堆排序

先堆化数组，再将堆顶A[0]与最后一个元素A[n-1]置换

再堆化数组，将堆顶A[0]与A[n-2]置换，重复，直到只剩下一个未排序元素

14.归并排序

将两个已排序的数组合并

但是它很难用于主存排序，主要问题在于需要线性附加内存，在整个算法中还要花费将数据拷贝到临时数组再拷贝回来的工作

15.快速排序

取一个枢纽元，将数组中大于和小于它的与元素分别进行快速排序，重复

16.递归

▲常见优化手段有：尾递归、迭代、循环

尾递归是指，在函数返回的时候，调用自身本身

并且，return语句不能包含表达式

这样，编译器或者解释器就可以把尾递归做优化，使递归本身无论调用多少次，都只占用一个栈帧，不会出现栈溢出的情况。 尾递归调用时，如果做了优化，栈不会增长，因此，无论多少次调用也不会导致栈溢出。 遗憾的是，大多数编程语言没有针对尾递归做优化

▲可以用非递归的方法实现二叉树前序遍历

▲程序调优的方法：

A缓存命中率提高，减少访问存储器的开销

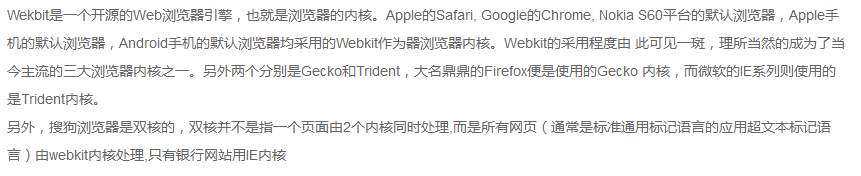
B多线程并不能使IO操作并行化，通过异步读写、合并读写、计算代替读写等方式优化

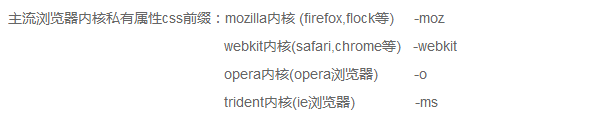
C可以减少连接和断开数据库的开销

D可以减少运行时对程序栈进行操作的开销

E减少远程调用的次数

17.浏览器





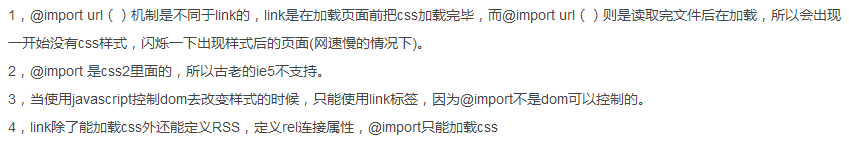


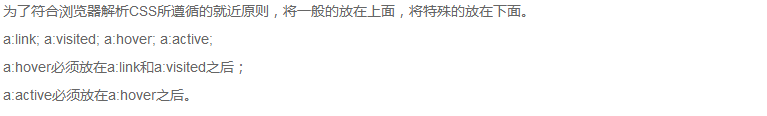
18.H5



19.前端







▲（1）只要是img src引用的都会请求；

（2）background: url()所属样式如果挂靠在某个元素上，就一定会请求；单独写了这么一个样式并不请求。

▲在html中，帧元素（frameset）的优先级最高，表单元素比非表单元素的优先级要高。

表单元素包括：文本输入框，密码输入框，单选框，复选框，文本输入域，列表框等等；

非表单元素包括：连接（a），div,table,span等。

所有的html元素又可以根据其显示分成两类：有窗口元素以及无窗口元素。有窗口元素总是显示在无窗口元素的前面。

有窗口元素包括：select元素，object元素，以及frames元素等等。

无窗口元素：大部分html元素都是无窗口元素。

▲（1）Ajax的优势：1.可搜索性 2.开放性 3.费用 4.易用性 5.易于开发

（2）Flash的优势：1.多媒体处理 2.兼容性 3.矢量图形 4.客户端资源调度

（3）Ajax的劣势：1.它可能破坏浏览器的后退功能   2.使用动态页面更新使得用户难于将某个特定的状态保存到收藏夹中 ，不过这些都有相关方法解决

（4）Flash的劣势：1.二进制格式 2.格式私有 3.flash 文件经常会很大，用户第一次使用的时候需要忍耐较长的等待时间

20.网络

▲局域网的网络地址192.168.1.0/24，局域网络连接其他网络的网关地址是192.168.1.1。主机192.168.1.20访问172.16.1.0/24网络时，其路由设置是：

route add -net 172.16.1.0 gw 192.168.1.1 netmask 255.255.255.0 metric 1

route add -net [目的子网] gw [网关地址] netmask [子网掩码] metric [路由跳数]

▲ping属于ICMP协议

21.数据库

▲Mysql复制分成三步：   
(1)master将改变记录到二进制日志(binary log)中（这些记录叫做二进制日志事件，binary log events）；   
(2)slave将master的binary log events拷贝到它的中继日志(relay log)；

(3)slave重做中继日志中的事件，将改变反映它自己的数据。

mysql支持master-slave复制，也支持master-master复制

一般情况下，异步复制的性能比半同步复制好，但后者相对更为安全

在有多个slave参与的半同步复制中，master并不一定需要等待全部slave返回

▲BEGIN或START TRANSACTION；显示地开启一个事务；

COMMIT；也可以使用COMMIT WORK，不过二者是等价的。COMMIT会提交事务，并使已对数据库进行的所有修改称为永久性的；

ROLLBACK；有可以使用ROLLBACK WORK，不过二者是等价的。回滚会结束用户的事务，并撤销正在进行的所有未提交的修改；

SAVEPOINT identifier；SAVEPOINT允许在事务中创建一个保存点，一个事务中可以有多个SAVEPOINT；

RELEASE SAVEPOINT identifier；删除一个事务的保存点，当没有指定的保存点时，执行该语句会抛出一个异常；

ROLLBACK TO identifier；把事务回滚到标记点；

SET TRANSACTION；用来设置事务的隔离级别。InnoDB存储引擎提供事务的隔离级别有READ UNCOMMITTED、READ COMMITTED、REPEATABLE READ和SERIALIZABLE。

# 七、前端面试题汇总- Html+H5

1. cookie/localStorage/sessionStorage/userData/session

（1）session由服务器存储数据，当客户端浏览器关闭，session释放

（2）cookie由浏览器本地存储小于4k（总量）的数据，但是需要传入服务器处理再传回浏览器

（3）localStorage html5新技术，由浏览器本地直接存储小于5M（总量）的数据，不需要经过服务器，存储格式为string，浏览器隐私模式下不可读取，localStorage本质上是对字符串的读取，如果存储内容多的话会消耗内存空间，会导致页面变卡，localStorage不能被爬虫抓取到

（4）**localStorage的写入**有三种方法

var storage= localStorage;

//写入a字段

storage["a"]=1;

//写入b字段

storage.a=1;

//写入c字段

storage.setItem("c",3);

读取：官方推荐的是getItem\setItem这两种方法对其进行存取

修改：storage.a=4;

清空：storage.clear();

删除：storage.removeItem("a");

对象的localStorage存储需要先通过JSON转化为字符串，JSON.stringify，提取则用JSON.parse

（5）localStorage与sessionStorage的唯一一点区别就是localStorage属于永久性存储，而sessionStorage属于当会话结束的时候，sessionStorage中的键值对会被清空

sessionStorage.lastname="Smith";

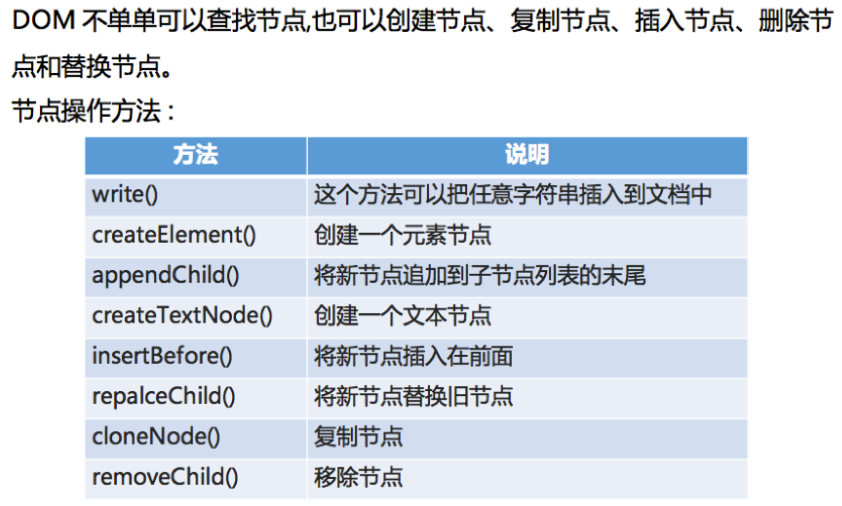
2.DOM结点增删查改

元素节点的 nodeName 是标签名称

属性节点的 nodeName 是属性名称

文本节点的 nodeName 永远是 #text

文档节点的 nodeName 永远是 #document



DOM操作是面试手写代码的重点

DOM元素获取和操作方法，新增元素，修改文本，事件代理，遍历元素，购物栏等题

都要能熟练手写出来

# 八、前端面试题汇总-CSS+CSS3

1.垂直居中方法

（1）单行

{

height:100px;

line-height:100px;

over-flower:hidden;

}

（2）不定高

{

padding-top:10px;

padding-bottom:10px;

}

（3）表格单元

{

display:table;  
}

{

display:table-cell;

height:50px;

text-align:center;

}

img{

vertical-align:center;  
}

（4）绝对定位

{

position:relative;

}

{

position:absolute;

top:50%;

margin-top:-25px;

}

或者

{

position:absolute;

top/bottom/left/right:0;

}

（5）flex

{

display:flex;

align-items;center;

justify-content:center;

}

（6）transform

{

position:absolute;

top:50%;

transform:translate(top,-50%);

}

2.flex弹性布局

<http://www.cnblogs.com/lixuemin/p/6110434.html>

3.CSS实现图片轮播

<style type="text/css">

\*{

margin:0;

padding:0;

}

#d{

margin:50px auto;

width: 600px;

height: 400px;

border:1px solid #9cc5ef;

overflow:hidden;

}

#u{

list-style: none;

height: 100%;

width: 300%;

animation:slide 3s infinite;

}

#u li{

float: left;

text-align: center;

line-height: 400px;

width: 600px;

height: 100%;

}

#u li:nth-child(1){

background: green;

}

#u li:nth-child(2){

background: blue;

}

#u li:nth-child(3){

background: red;

}

@keyframes slide {

0% { margin-left:0;}

23% { margin-left:0;}

33% { margin-left:-600px;}

56% { margin-left:-600px;}

66% { margin-left:-1200px;}

90% { margin-left:-1200px;}

100% {margin-left:0;}

}

</style>

<div id="d">

<ul id="u">

<li>1</li>

<li>2</li>

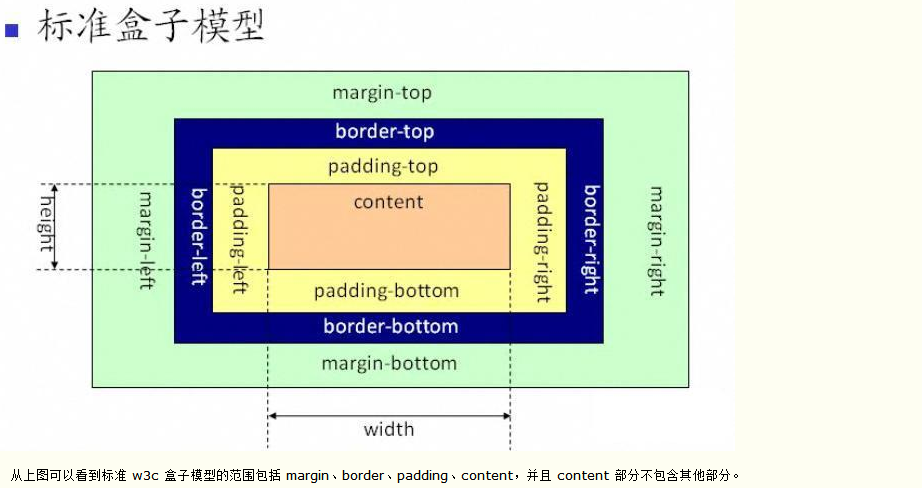
<li>3</li>

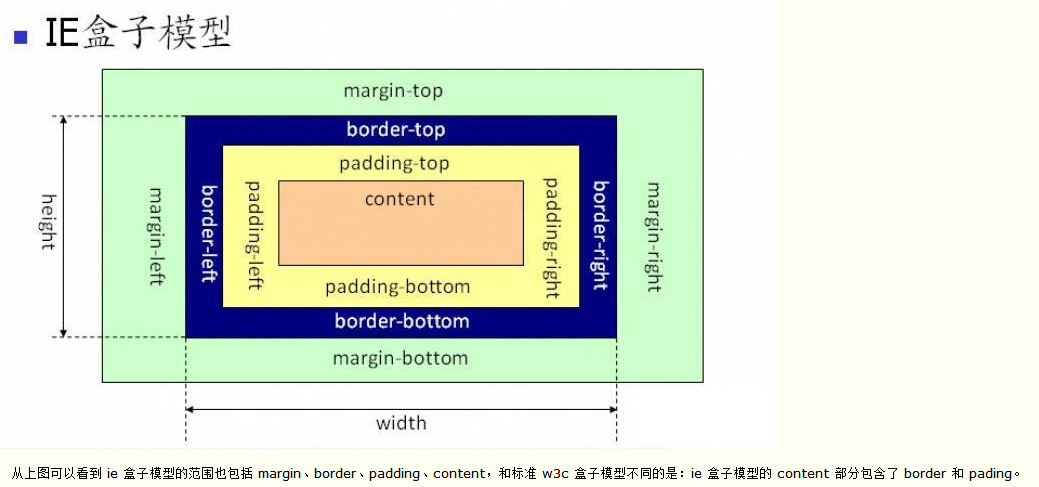
</ul>

</div>

方法有很多，百度查，耐心看，动手敲，理解原理

4.盒子模型





5.display:none;和visibility:hidden和overflow:hidden区别

display:none的元素不会被加载，在页面中没有空间位置，元素彻底消失

overflow:hidden的元素会把超出设置宽高的部分剪掉，多余部分虽然加载了，但是不会占空间位置

visibility:hidden的元素仍会被加载，只是显示空白，页面上占有空间位置

6.BFC

Block Formatting Context块级格式化范围

HTML元素在这个环境中按照一定规则进行布局，产生BFC的容器内部独立布局，与外部不相干

如何产生BFC：当一个HTML元素满足下面条件的任何一点，都可以产生Block Formatting Context：

float的值不为none。

overflow的值不为visible。

display的值为table-cell, table-caption, inline-block中的任何一个。

position的值不为relative和static。

<http://www.cnblogs.com/nujufoul/p/7092520.html>

7.overflow属性

inherit继承父元素的overflow属性

visible默认值，内容不会被修剪，超出容器的元素会显示在容器外

hidden内容会被修剪，超出容器的元素不可见且不占位置

scroll内容会被修剪，浏览器会始终显示滚动条，可查看超出部分

auto如果内容被修剪，会显示滚动条显示超出部分

8.边框重叠

两个块级元素垂直方向margin-top，margin-bottom会重叠，取较大的margin值作为公共外边距

9.三列布局

<http://www.cnblogs.com/honoka/p/5161836.html>

10.float

float浮动，在父容器中向上浮动，仍然在文档流中

前面有block元素则停止，inline元素向后重构

浮动元素对于block元素没有体积，会被覆盖

# 九、前端面试题汇总-JS+ES6

1.JS返回类型的方法

typeof：识别基础类型，funtion可以输出”function”，null会输出”object”

Object.prototype.toString.apply()：在ie6/7/8会把null和undefined识别为”[object Object]”

instanceof：在不同iframe和window之间失效，返回true/false

isArray：Array对象的方法

constructor：返回对象的构造函数

2.instanceof原理

A instanceof B检测一个对象A是不是另一个对象B的实例的原理是：查看对象B的prototype对象是否在对象A的[[prototype]]（原型）链上。如果在，则返回true,如果不在则返回false。不过有一个特殊的情况，当对象B的prototype为null将会报错(类似于空指针异常)

3.js设计模式

<http://www.cnblogs.com/tugenhua0707/p/5198407.html#_labe0>

4. JS事件onchange、oninput、onpropertychange区别

onchange事件在输入框内容改变且失去焦点时触发

oninput 事件不支持ie9以下，value改变时会实时触发，通过js改变value时不会触发，从浏览器的下拉提示中选取时不会触发，oninput要通过addEventListener()来注册

onpropertychange事件为ie专属，任意属性改变都会实时触发，通过js改变也会触发

5.this指向

（1）全局this和普通函数中的this表示window

（2） 对象方法中的this表示调用该方法的对象，普通函数引用对象方法，则this表示window，因为调用方法的并不是对象，而是函数

（3）对象方法中独立函数（如setTimeout中的匿名函数）的this表示window

（4）构造器中的this表示新创建的实例对象

（5）DOM事件绑定函数中的this表示该DOM元素

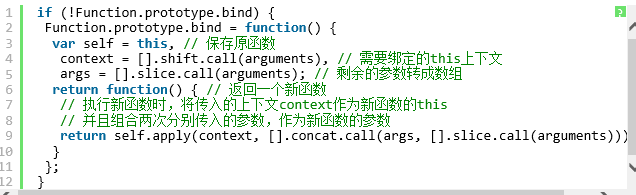
（6）箭头函数中的this在定义时就已经绑定，与上下文中this指向相同

6.bind实现原理

call，apply临时绑定this，直接调用函数

bind生成一个与原函数一样的绑定this和实参函数，返回这个新函数

当bind()所返回的函数用作构造函数的时候，传入bind()的this将被忽略，但实参会全部传入原函数



7.new一个实例，中间经过哪几个过程

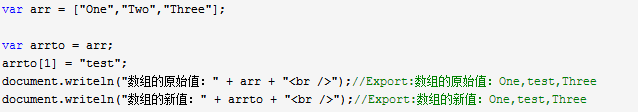
（1）建立一个新对象

（2）将新对象的\_\_proto\_\_指向构造函数的prototype

（3）绑定构造函数的this为新对象，执行构造函数，返回新对象

8.深/浅拷贝

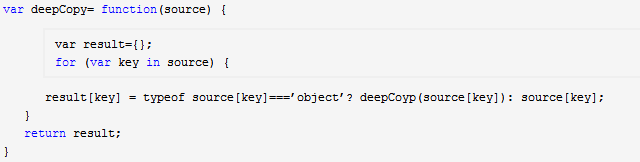
数组浅拷贝：直接将数组名赋值给其他变量时，赋给该变量的只是一个对象的引用，当改变数组时，两个变量也将改变



数组深拷贝：通过方法对数组操作，生成新的数组对象再赋值给一个变量

如：arr.slice(0); arr.concat();

对象深拷贝：建立一个新的对象，将原对象属性都赋给新对象



9.原型链

需要理解，自己找资料补充

10.闭包

<http://www.jb51.net/article/24101.htm>

11.作用域链

需要理解，自己找资料补充

12.Ajax原理

判断是否支持XMLHttpRequest，有则创建一个新的XMLHttpRequest对象，ie5/6不支持则创建一个新的ActiveXObject对象

if ( window.XMLHttpRequest ){

xmlhttp = new XMLHttpRequest();

}else{

xmlhttp = new ActiveXObject(“Microsoft.XMLHTTP”);

}

发送Ajax请求

xmlhttp.open(“GFT/POST”,url,true/false);

xmlhttp.send();

Ajax响应

xmlhttp.onreadystatechange = function(){

if (xmlhttp.readystate==4 && xmlhttp.status==200){

xmlhttp.responseText/xmlhttp.responseXML  
}

}

13.js内存泄漏的操作

内存泄露是指一块被分配的内存既不能使用，又不能回收，直到浏览器进程结束

（1）当页面中元素被移除或替换时，若元素绑定的事件仍没被移除，在IE中不会作出恰当处理，此时要先手工移除事件，不然会存在内存泄露

（2）在 Internet Explorer 中，如果循环引用中的任何对象是 DOM 节点或者 ActiveX 对象，垃圾收集系统则不会发现它们之间的循环关系与系统中的其他对象是隔离的并释放它们。最终它们将被保留在内存中，直到浏览器关闭

（3）闭包可以维持函数内局部变量，使其得不到释放。

解决之道：

减少使用闭包

或者将闭包作用域中没用的属性删除，以减少内存消耗

（4）删除对象，如果对象的属性被引用

在JavaScript的某些实现中，在销毁对象的时候，要遍历属性中属性，依次删除

14.setTimeout和setInterval原理

setTimeout(func,time)：

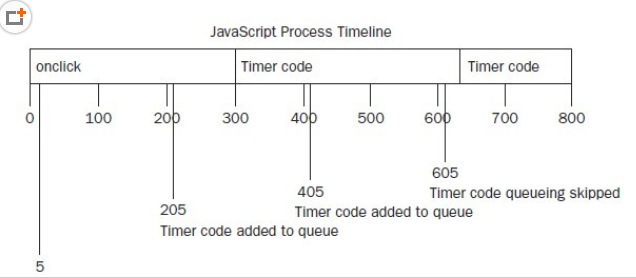
在时间time后将func代码段插入到js代码队列中，如果time时间在执行另一个代码段，则需要等这个代码段执行完毕再执行func代码段

setInterval(func,time)：

（1）有可能跳过时间间隔

（2）时间间隔可能小于time

同样在time时间后，js代码队列插入func代码段，等正在的程序执行完之后，执行func代码段，time是从开始计时的，如果func执行时间过长，2\*time时间后又一个func插入队列，如果3\*time时间后第2个func还未执行，那么第3个func会被跳过，因为js引擎只允许一个未执行的func



一种解决方案：

setTimeout(function(){   
//processing   
setTimeout(arguments.callee/\*表示当前正在执行的函数\*/, interval);   
}, interval);

当func程序执行完后再setTimeout

15.js改变css样式

obj = document.getElementById(“btn”)

（1）修改单个嵌入式样式

obj.style.textAlign = “center”;

如果属性有'-'号，就写成驼峰的形式（如textAlign）  如果想保留 - 号，就中括号的形式  element.style['text-align'] = 'center';

obj.setAttribute('height', '100px');

（2）修改整体嵌入式样式

obj.style.cssText = “background-color:black; display:block; color:White;”

obj.setAttribute('style', 'height: 100px !important');

（3）修改class名

obj.className = “btnA”;

obj.setAttribute(“class”,”btnA”);

（4）更改使用的外联css文件

<link href="css1.css" rel="stylesheet" type="text/css" id="css"/>

function changeStyle4() {

   var obj = document.getElementById("css");

   obj.setAttribute("href","css2.css");

 }

（5）使用setProperty

如果要设置!important，推荐用这种方法设置第三个参数

element.style.setProperty('height', '300px', 'important');

16.js继承

（1）原型链继承

将父类实例赋值给子类的原型prototype

（2）构造继承

在子类构造函数中执行父类构造函数，相当于把父类的实例属性方法复制给子类实例

（3）实例继承

子类构造函数其实相当于一个普通函数，子类构造函数中生成一个父类实例，然后在父类实例上添加属性方法，返回父类实例，子类实例仍然在父类原型链上

（4）拷贝继承

与实例类似，不同的是生成父类实例后，对父类实例的属性枚举，将其赋值给子类的原型prototype，

再添加子类自己的prototype属性，子类实例在子类的原型链上

（5）组合继承

结合原型链继承和构造继承

详细信息：

<https://m.baidu.com/from=1000953f/bd_page_type=1/ssid=6e30c4abc6e2b2d4f1b7ea29/uid=0/pu=usm%402%2Csz%40320_1001%2Cta%40iphone_2_7.0_3_537/baiduid=6D3B273E7E68C3318E51A2804A8F7FAA/w=0_10_/t=iphone/l=3/tc?ref=www_iphone&lid=9064722612923587903&order=1&fm=alop&tj=www_normal_1_0_10_title&vit=osres&m=8&srd=1&cltj=cloud_title&asres=1&title=JS%E5%AE%9E%E7%8E%B0%E7%BB%A7%E6%89%BF%E7%9A%84%E5%87%A0%E7%A7%8D%E6%96%B9%E5%BC%8F-%E5%B9%BB%E5%A4%A9%E8%8A%92-%E5%8D%9A%E5%AE%A2%E5%9B%AD&dict=30&w_qd=IlPT2AEptyoA_yivDsV78_LV1IIhliVJ-pJWi_eiurxiQa&sec=23945&di=df3da9dbd3a36a0d&bdenc=1&tch=124.0.261.246.0.0&nsrc=IlPT2AEptyoA_yixCFOxXnANedT62v3IEQGG_ytK1DK6mlrte4viZQRAWiLyKXXTUS4dgTCctRoIxnyd08hunM5X&eqid=7dcc5d3082c2dc001000000059b9677b&wd=&clk_info=%7B%22srcid%22%3A%221599%22%2C%22tplname%22%3A%22www_normal%22%2C%22t%22%3A1505322877172%2C%22sig%22%3A%221804%22%2C%22xpath%22%3A%22div-div-div-a-p%22%7D&sfOpen=1>

18.js设计模式

<http://www.cnblogs.com/tugenhua0707/p/5198407.html#_labe0>

# 十、前端面试题汇总-专业基础知识

1.域名从输入到响应过程

（1）域名解析

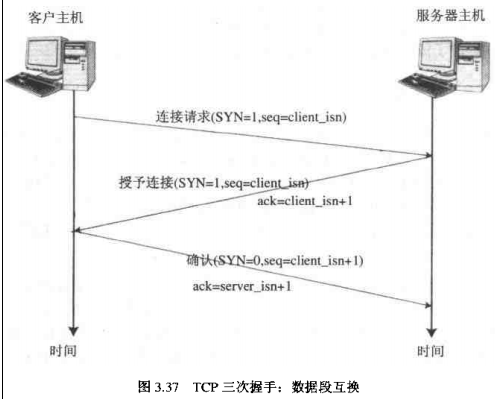
查询浏览器自身DNS缓存

查询操作系统自身DNS缓存

尝试读取hosts文件

向本地配置首选DNS服务器发送请求

（2）与服务器建立通信（TCP三次握手）



（3）发送http请求

建立TCP连接后浏览器向服务器发送http请求

浏览器发送其请求命令之后，还要以头信息的形式向Web服务器发送一些别的信息

之后浏览器发送了一空白行来通知服务器，它已经结束了该头信息的发送

（4）服务器处理请求

服务器收到http请求，确定执行什么(ASP.net PHP RUBY JAVA等)来处理它，读取参数并进行逻辑操作后，生成指定的数据

（5）服务器做出应答

客户机向服务器发出请求后，服务器会客户机回送应答，HTTP/1.1 200 OK ，应答的第一部分是协议的版本号和应答状态码

（6）服务器发送应答头信息

正如客户端会随同请求发送关于自身的信息一样，服务器也会随同应答向用户发送关于它自己的数据及被请求的文档

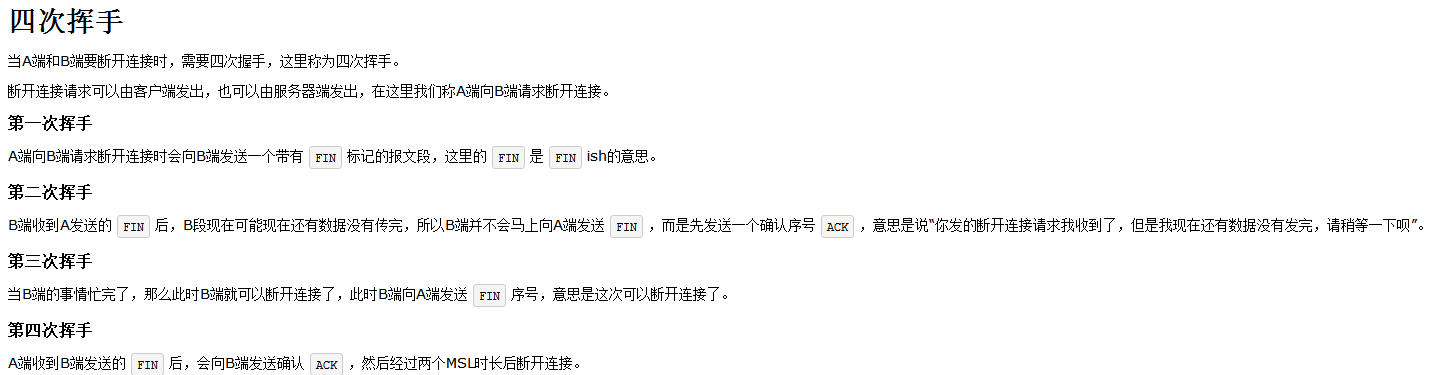
（7）服务器发送数据

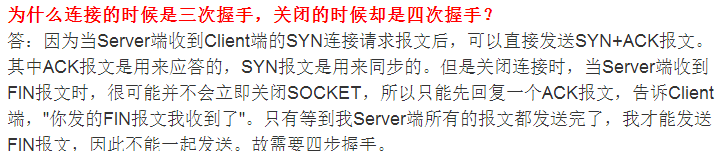
Web服务器向浏览器发送头信息后，它会发送一个空白行来表示头信息的发送到此为结束，接着，它就以Content-Type应答头信息所描述的格式发送用户所请求的实际数据

（7）tcp连接关闭

一般情况下，一旦Web服务器向浏览器发送了请求数据，它就要关闭TCP连接（四次挥手）

然后如果浏览器或者服务器在其头信息加入了这行代码：Connection:keep-alive   
TCP连接在发送后将仍然保持打开状态，于是，浏览器可以继续通过相同的连接发送请求。保持连接节省了为每个请求建立新连接所需的时间，还节约了网络带宽



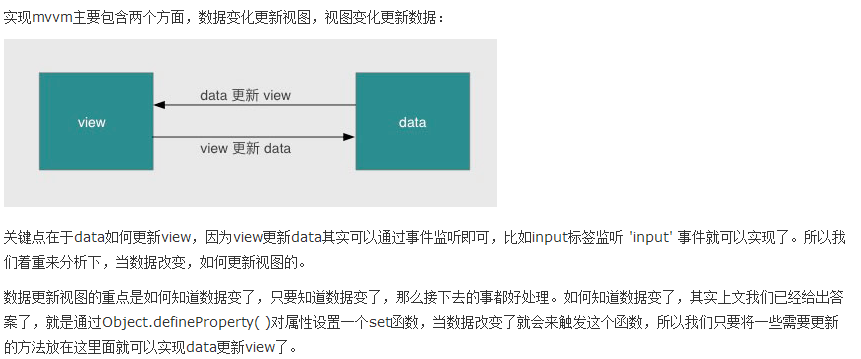


当tcp第三次握手失败，服务器不会重传报文，而是发送RTS报文段，进入closed状态，防止syn洪泛攻击

2.进程和线程区别

<http://blog.csdn.net/yaosiming2011/article/details/44280797>

3.mvvm数据双向绑定原理



<http://www.cnblogs.com/libin-1/p/6893712.html>

4.前端优化规则

<http://blog.csdn.net/lysunnyrain/article/details/51068508>

5.内联图片

<img src=”data:image/png;base64, iVBOR....”/>

image/png - 数据类型名称，base64 - 数据的编码方法，iUANR.... - 编码后的数据

: , ; - data URI scheme 指定的分隔符号

优点：

（1）减少http请求次数

（2）做为背景平铺类的图片使用内联图片的话，减少http请求次数，并且不会影响加载速度

**缺点：**

（1）浏览器不会缓存内联图片资源

（2）兼容性较差，只支持ie8以上浏览器

（3）超过1000kb的图片，base64编码会使图片大小增大，导致网页整体下载速度减慢

如何获取图片的base64编码呢。网络上有很多免费的base编码和解码工具，但是有个最简单方法就是我们写一个PHP文件。使用base64\_encode()进行编码：比如：

echo base64\_encode(file\_get\_contents('211-11.JPG'));

6.浏览器如何完成渲染

（1）IE下载的顺序是从上到下，渲染的顺序也是从上到下，下载和渲染是同时进行的。  
（2）在渲染到页面的某一部分时，其上面的所有部分都已经下载完成（并不是说所有相关联的元素都已经下载完）。  
（3）如果遇到语义解释性的标签嵌入文件（JS脚本）会启用单独连接进行下载。  
（4）并且在下载后进行解析，解析过程中，停止页面所有往下元素的下载。  
（5）样式表在下载完成后，将和以前下载的所有样式表一起进行解析，解析完成后，将对此前所有元素（含以前已经渲染的）重新进行渲染。  
（6）JS、CSS中如有重定义，后定义函数将覆盖前定义函数

<http://blog.csdn.net/liaozhongping/article/details/51028705>

7.前后端分离

前后端不分离：html+css链接+js链接+后端语言混合开发，后端先执行后端语言，从数据提取数据，渲染页面，一起发送给浏览器

前后端分离：只发送html+css链接+js链接给前端，通过Ajax和后端接口（路由、控制器）返回所需的数据，再通过数据绑定布局到页面中

前后端不分离：

优点：返回页面时同时返回了该页面所需的数据，不会出现数据延迟或数据失败的情况

缺点：前后端开发效率低，维护不方便，不适用于开发复杂的业务

前后端分离：

优点：开发维护方便，前后端开发人员可以专注于自己的部分

缺点：增加沟通成本，需要前后端联调测试，有数据延迟，可能数据丢失

8.跨域的方式有几种

（1）代理

因为同源策略是针对客户端的，在服务器端没有什么同源策略，是可以随便访问的，所以我们可以通过下面的方法绕过客户端的同源策略的限制：

客户端先访问同源的服务端代码，该同源的服务端代码

使用httpclient等方法，再去访问不同源的服务端代码，然后将结果返回给客户端，这样就间接实现了跨域。

相关例子，参见博文：http://www.cnblogs.com/digdeep/p/4198643.html

（2）cors

在服务端开启cors也可以支持浏览器的跨域访问

cors即：Cross-Origin Resource Sharing 跨域资源共享

jsonp和cors的区别是jsonp几乎所有浏览器都支持，但是只能是get

而cors有些老浏览器不支持，但是get/post都支持

**（3）PHP端修改header（XHR2方式）**

在php接口脚本中加入以下两句即可：

header('Access-Control-Allow-Origin:\*');//允许所有来源访问

header('Access-Control-Allow-Method:POST,GET');//允许访问的方式

（4）JSONP

9.JSONP原理

ajax请求受浏览器同源策略影响，不允许进行跨域请求，而script标签src属性中的链接却可以访问跨域的js脚本，利用这个特性，服务端不再返回JSON格式的数据，而是返回一段调用某个函数的js代码，在src中进行了调用，这样实现了跨域

<http://blog.csdn.net/u011897301/article/details/52679486>

10.webpack脚手架

# 十一、前端面试题汇总-框架

1.BootStracp栅格布局



col-xs- col-sm- col-md- col-lg-

具体看官网

2.Vue生命周期&钩子函数

beforeCreate 组件实例刚被创建

created 组件实例创建完成，属性已绑定，DOM还未完成，$el属性还不存在

beforeMount 模板编译之前

mounted 模板编译之后

beforeUpdate 组件更新之前

updated 组件更新之后

activated 组件被激活时调用

deactivated 组件被移除时调用

beforeDestory 组件销毁前调用

destoryed 组件销毁后调用

<https://segmentfault.com/a/1190000008010666>

2.Vue组件通信

<http://www.cnblogs.com/QRL909109/p/6166209.html>

<http://blog.csdn.net/qq_24122593/article/details/53010758>

3.Vue双向数据绑定

<https://segmentfault.com/a/1190000006599500>

4.项目

项目架构、结构图、难点、亮点

自己的项目要非常熟悉，可以现场对项目进行优化、修改需求

# 十二、前端面试题汇总-网络

1.Socket通信

本地通过进程PID唯一标识进程

网络中的进程通过“ip+协议+端口”唯一标识进程

网络进程之间通信使用：UNIX BSD套接字（socket）

socket是在应用层和传输层之间的一个抽象层，它把TCP/IP层复杂的操作抽象为几个简单的接口供应用层调用，以实现进程在网络中通信

socket(地址类型，socket类型，协议)、bind(服务器地址，端口号)、listen、connect、accept、read、write、close

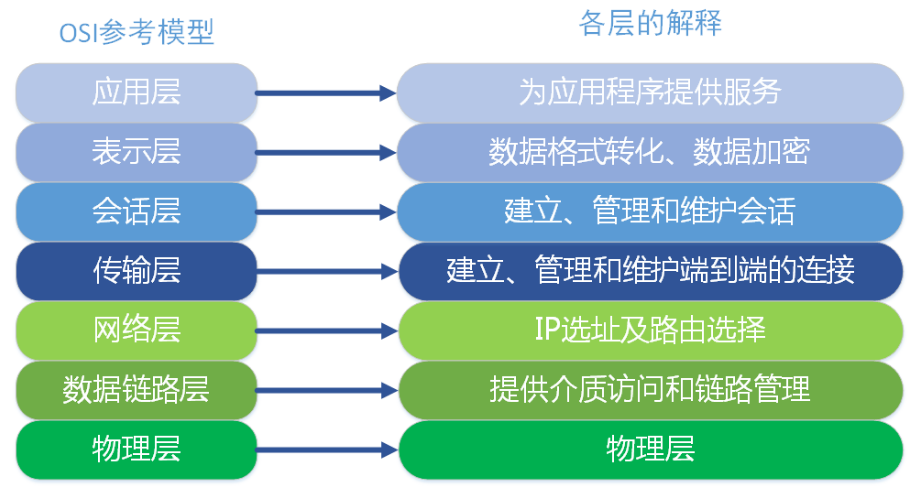
socket类型：

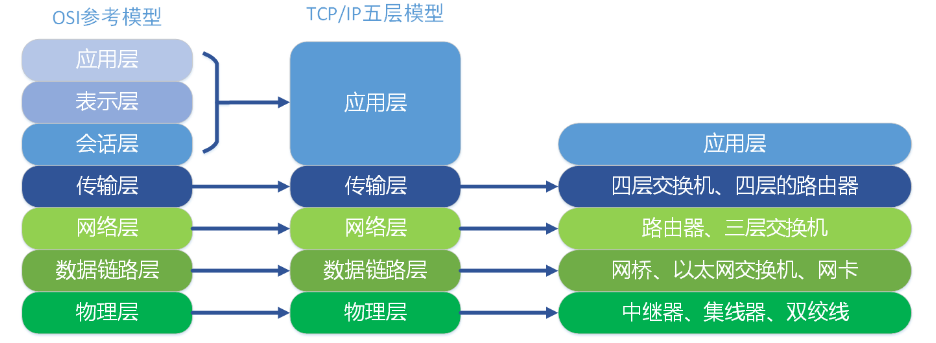
  (1)基于TCP的Socket:提供给应用层可靠的流式数据服务，使用TCP的Socket应用程序协议：BGP，HTTP，FTP，TELNET等。优点：基于数据传输的可靠性。  
   (2)基于UDP的Socket：适用于数据传输可靠性要求不高的场合。基于UDP的Socket应用程序或协议有：RIP，SNMP，L2TP等。  
  (3)基于RawIp的Socket:非连接，不可靠的数据传输。特点：能使应用程序直接访问网络层。基于RawIp的Socket有ping ,tracert,ospf等。  
  (4)基于链路层的Socket。为IS-IS协议提供的Socket接口。使IS-IS协议可通过Socket直接访问链路层。非连接，不可靠通信服务。

<http://www.cnblogs.com/dolphinX/p/3460545.html>

<http://blog.csdn.net/jiajia4336/article/details/8798421>

2.OSI七层模型和TCP/IP五层模型





TCP/IP四层模型：应用、传输、互联网络、网络接口

3.TCP/IP协议

应用层：

tcp：

SSH，FTP（文件传输协议），HTTP（超文本传输协议），SMTP（邮件发送协议），POP3（邮件接收协议），Telnet （远程控制协议）

udp：

TFTP（简单文件传输），SNMP（简单网络管理协议）

tcp&udp：

DNS（域名解析）

icmp：

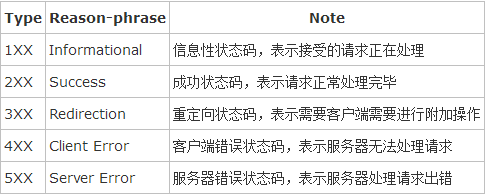
ping

传输层：TCP，UDP

网络层：IP，ICMP，OSPF，EIGRP，IGMP

数据链路层：SLIP，CSLIP，PPP，MTU

4.Http状态码



**常用状态码：**

**200 OK**  
  一切正常，对GET和POST请求的应答文档跟在后面

**301 Moved Permanently**  
  客户请求的文档在其他地方，新的URL在Location头中给出，浏览器应该自动地访问新的URL。

**302 Found**  
  类似于301，但新的URL应该被视为临时性的替代，而不是永久性的

**304 Not Modified**  
  客户端有缓冲的文档并发出了一个条件性的请求（一般是提供If-Modified-Since头表示客户只想比指定日期更新的文档）。服务器告诉客户，原来缓冲的文档还可以继续使用。

**307 Temporary Redirect**  
  和302（Found）相同。许多浏览器会错误地响应302应答进行重定向，即使原来的请求是 POST，即使它实际上只能在POST请求的应答是303时才能重定向。由于这个原因，HTTP 1.1新增了307，以便更加清楚的区分几个状态代码： 当出现303应答时，浏览器可以跟随重定向的GET和POST请求；如果是307应答，则浏览器只能跟随对GET请求的重定向

**400 Bad Request**  
  请求出现语法错误。

**401 Unauthorized**  
  客户试图未经授权访问受密码保护的页面。应答中会包含一个WWW-Authenticate头，浏览器据此显示用户名字/密码对话框，然后在填写合适的Authorization头后再次发出请求

**403 Forbidden**  
  资源不可用

**404 Not Found**  
  无法找到指定位置的资源

**410 Gone**  
  所请求的文档已经不再可用，而且服务器不知道应该重定向到哪一个地址

**500 Internal Server Error**  
  服务器遇到了意料不到的情况，不能完成客户的请求

**501 Not Implemented**  
  服务器不支持实现请求所需要的功能。例如，客户发出了一个服务器不支持的PUT请求

5.TCP和UDP区别

TCP：

面向连接，三次握手

20字节报头，结构复杂，流模式

数据正确，保证顺序

UDP：

面向报文，传送速度取决于应用程序产生数据速度、计算机能力、传输带宽，接收方将数据放在队列中，每次读取一个消息段

8字节报头，结构简单，数据报模式

不保证数据正确，不保证顺序正确

6.Http头信息

<http://blog.csdn.net/guoguo1980/article/details/2649658>

7.https加密过程

<http://www.cnblogs.com/zery/p/5164795.html>