前端面试宝典

既然求了雨，就要承受泥泞

班级：SC201201Web  
姓名：

Git专题

1. 如何创建仓库，添加一个版本？（重要）

git init  
git add -A  
git commit -m “first”

1. Git和SVN的区别是？（必须）

Git 是分布式的，SVN 不是

Git 把内容按元数据方式存储，而 SVN 是按文件

Git 分支和 SVN 的分支不同

Git 没有一个全局的版本号，而 SVN 有

Git 的内容完整性要优于 SVN

1. 文件工作区被修改，没有添加到暂存盘，请问如何撤销？  
     
   文件没有被关闭的情况下，可以使用软件自带恢复 ctrl+z 或者 git checkout -- readme.txt
2. 文件在工作区被修改，并且已经添加到暂存盘，请问如何撤销？

git reset HEAD readme.txt  
git checkout -- readme.txt

1. 如何恢复到当前版本的上一个版本？（重要）

git reset --hard HEAD^

1. 如何恢复到指定的版本？（重要）

git reset --hard 版本号

1. 请介绍一下如何生成 SSH KEY？

再命令窗口输入以下代码

ssh-keygen -t rsa -C "youremail@example.com"

然后默认按回车，如果是第二次生成，会提示覆盖，你选择yes即可

1. 请介绍一下如何添加远程仓库，那些字段是可以自定义的？（必须）

git remote add 远程仓库名 远程仓库地址

远程仓库名 是自定义的 一般github 使用 origin

1. 请介绍一下如何推送本地仓库到远程仓库？（必须）

git add -A

git commit -m “描述文字”

git remote add origin 仓库地址

git push -u origin master

1. 分支创建切换的步骤是？

git switch -c 分支名

git checkout -b 分支名

git branch 分支名

git switch 分支名

1. 如何强制删除分支？

git branch -D 分支名

1. 不能git pull，如何解决？（重要）

有以下可能：

1. 前工作区做了修改，并没有提交一个版本   
   b. 需要缓存一下当前工作区 git stash  
   c. 本地分支和远程分支没有建立关联  
   d. 两个不同的版本库，可以强制合并（git pull origin 分支名 --allow-unrelated-histories）
2. 抓取分支的具体步骤是？

git switch -c 分支名 origin/dev

git checkout -b 分支名 origin/dev

1. .gitignore是什么文件？

有时候我们并不想把一些敏感信息、设计稿源文件或者缓存文件上传到远程仓库，那么我们可以通过.gitignore文件设置一个忽略上传文件名单

1. https和ssh协议区别? （重要）

https协议在推送的时候会弹出登录框，输入仓库托管网站（gitee或者github等）的账户和密码，输入成功后，我们的操作系统会自动记住登录状态，下一次推送不需要再输入账号和密码

ssh协议需要自己生成公钥，并且和仓库托管网站登录账户进行绑定，即可。绑定成功后，也是可以免密码登录推送的。

1. 请介绍一下远程协作的步骤 （重要）

假设有一个现存项目vivocms

1. 申请成为开发者

2. 得到仓库地址，克隆项目（SSH协议，无登陆提交）

3. 注意克隆到本地的项目会默认建立和远程仓库的绑定，默认仓库名origin

4. 开始编写代码，添加版本，继续写代码、添加版本.....

5. 开始提交项目到远程 git push -u origin master

6. 发送冲突，因为同事的比你先提交一个版本。

7. 使用git pull拉取最新的版本信息，拉取过程可能会产生冲突，手动解决冲突。

8. 解决好冲突保存版本git commit

9. 开始提交到远程git push，可能又会冲突，原因是又一个同事比你早一个版本。

10.接下来重复6-9这个步骤，一直到push成功，你的版本才被正式提交到远程。

1. 两个不同独立仓库git pull拉取操作

1. gitee码云创建一个空仓库oa2020，注意勾选readme文件

2. 一旦勾选了readme，那么服务器会使用git init初始化一个远程仓库，并且提交第一次版本

3. 在本地创建oa2020目录，使用git init初始化仓库，那么注意此时本地oa2020 和远程 gitee  
 oa2020是各自独立的仓库。

4. 本地仓库添加一个版本，然后使用git push到远程仓库会出错，原因是因为远程仓库已经有一次  
提交版本，你必须拉取远程仓库的版本，不然会因为冲突无法提交。

5. 那么你会想到使用git pull，但是发现git pull 无法拉取远程仓库版本信息，因为是本地和远程仓  
库是不同的独立仓库，要拉取必须使用 --allow-unrelated-histories

例如 git pull gitee master --allow-unrelated-histories

6. 注意如果你是使用git clone 远程仓库，那么肯定不是各自独立的仓库，不需要使用  
--allow-unrelated-histories属性

1. git log 和 git reflog的区别？

git log查看历史版本到当前版本的日志  
git reflog查看所有历史版本信息，注意这里查看的是一个短的版本号，例如 da213db

1. git diff 是什么指令？

显示暂存区和工作区的差异

git diff --cached [file] 或者git diff --staged [file]

显示暂存区和上一次提交(commit)的差异

HTML专题

1. html5有什么新特性？（必须）

新元素  
 新增表单控件和语义化标签

新属性  
autocomplete 自动完成、required 必填、pattern 正则表达式和placeholder 提示占位文字

完全支持 CSS3

CSS圆角、盒阴影、文字阴影、颜色 hsl 和 rgba、媒体查询、webfonts和CSS动画等

video 和 audio

2D/3D 制图

svg和canvas制图

Web 应用

本地数据存储、本地 SQL 数据、离线存储、worker 多线程和xmlHttpRequest 2

1. 如何区分HTML和HTML5？

DOCTYPE声明新增的结构元素、功能元素

1. 主流浏览器以及引擎有那些？（必须）

chrome 谷歌 , 引擎以前是Webkit，现在是Blink内核  
firefox 火狐，引擎是 Gecko 内核

ie 微软（测试兼容性 IE9-），引擎是Trident内核（IE、360、搜狗）

opera 欧鹏，引擎以前是Presto内核，后来是Webkit内核，现在是Blink内核

safari 苹果，引擎是 Webkit内核

360 安全，引擎是IE+Chrome双内核，Trident（兼容模式），+Webkit（高速模式）

1. 前端常见web编辑器有那些？(了解)

Webstorm 前端开发神器  
VSCode 开源免费好用  
Sublime Text3 非常优雅的编辑器  
HBuilderX 极客开发工具  
Editplus 速度很快  
vim linux下默认编辑器

1. 空元素和单标签有那些？（重要）

空元素指没有内容的元素

例如：br 换行 和hr 水平线

单标签是指没有结束标签的元素  
例如：img图像和input输入框

1. 标签公共属性有那些？

id 定义唯一标识

class 定义类名

name 定义名称

style 定义行内样式

title 定义划过气泡提示

1. html和xhtml有什么区别？（必须）

最主要的不同：  
XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

标签名必须用小写字母。

XHTML 文档必须拥有根元素。

1. xml和 json的区别？（重要）

xml: 可扩展标记语言，是一种用于标记电子文件使其具有结构性的标记语言。

json：一种轻量级的数据交换格式。

两者区别  
它们都是一种数据交换格式。

1、xml是重量级的，json是轻量级的。

2、xml在传输过程中比较占带宽，json占带宽少，易于压缩。

3、xml和json都用在项目交互下，xml多用于做配置文件，json用于数据交互。

1. SEO优化有那些方面？（必须）
2. 设置窗口标题
3. 设置meta描述
4. 设置meta关键字，多个关键字用英文逗号分割
5. 一个页面只用一个h1，要合理的使用标题级别
6. 内链和外链优化
7. 软文推广，就是写一篇看似很有营养的文章，这些文章被大流量网站收录，文章里面植入了你网站  
    链接，属于外链优化
8. 代码要语义化和代码结构简练
9. 少用图片、视频、音频，因为搜索引擎无法采集图片、视频等富媒体上的内容
10. 常见的文本格式化标签有那些？（重要）

b 加粗 i 倾斜 s 中划线 small 小号文本 big 大号文本 sup 上标 sup 下标

em 倾斜有语义 strong加粗有语义 pre预处理文本 mark高亮 del下划线表示删除

ins 下划线表示插入等

1. 块级标签、行级标签和行块标签之间的区别？（必须）

块级标签特点：

1. 独占一行

2. 宽度是自适应的

3. 具有盒子属性（宽、高、边框、边距、填充）

内联（行级）标签特点：

1. 内联标签之间可以并排

2. 不能设置宽高

3. 可以设置边框、外边距（仅支持左和右）

内填充支持，但是渲染有问题，由此可见内

联再盒子默写渲染方案存在问题，不准确。  
  
行块级标签特点：  
1. 具有块级标签的特点，比如可以很好的渲染盒子属性，又具有内联标签的特点，比如并排。  
2. 行块之间有间隙

1. html5新增那些表单标签？（必须）

date 日期  
color 颜色  
number 数值  
tel 电话  
search 搜索  
url 网址  
time 时间  
email 邮箱  
datalist 预定义列表  
range 滑块

1. html字符常见实体有那些？（重要）

空格 &nbsp;

< 左尖括号 &lt;

> 右尖括号 &gt;

& 和号 &amp;

" 双引号 &quot;

' 单引号 &#39;

© 版权 &copy;

® 注册商标 &reg;

1. 一个标准的URL地址包含那些部分？（必须）

scheme 协议部分

host 主机部分

domain 域名部分

:port 端口部分

path 路径部分

filename 文件名部分

1. 常见兼容性问题有那些?（重要）

png24位的图片在IE6浏览器上出现背景  
解决方案：做成PNG8

浏览器默认的 margin 和 padding 不同  
解决方案：加一个全局的\*{margin:0;padding:0;}来统一  
  
任何浏览器都有默认字号，IE6的默认字号是14px，所有小于字号的盒子都不能正常渲染，高度不能小于字号。  
解决方案：就是用\_单独给IE6设置一个盒子的字号，字号比height小就行了，一般来说是0

连续多个元素浮动，浮动的方向和margin的方向相同，最后和最开头有可能出现双倍的margin。

IE6不支持position:fixed

IE6、7、8中，所有有超级链接的图片，都默认有一个蓝色边框  
解决方案：a img{border:none;}

opacity就是透明度的意思IE6~8不支持, 要写IE自己的属性  
解决方案：div{ filter:alpha(opacity=40) }

1. 解释在ie低版本下的怪异盒模型和c3的怪异盒模型 和 弹性盒模型?

IE当padding+border的值小于width或者height

盒模型的宽度=margin(左右)+width（width已经包含了padding和border的值）

盒模型的高度=margin(上下)+height（height已经包含了padding和border的值）

当padding+border的值大于width或者height

盒模型的宽度=margin(左右)+padding(左右)+border(左右)

盒模型的高度=margin(上下)+padding(上下)+border(上下)+19px （加一个默认行高 19px） 所以相当于是padding+border和width或者height比大小，谁大取谁。

以上几种DOCTYPE都是标准的文档类型，无论使用哪种模式完整定义DOCTYPE，都会触发标准模式，而如果DOCTYPE缺失则在ie6，ie7，ie8下将会触发怪异模式（quirks 模式） CSS3box-sizing有两个值一个是content-box，另一个是border-box。

当设置为box-sizing:content-box时，将采用标准模式解析计算，也是默认模式；

当设置为box-sizing:border-box时，将采用怪异模式解析计算；

Css3弹性盒模型引入了新的盒子模型—弹性盒模型，该模型决定一个盒子在其他盒子中的分布方式以及如何处理可用的空间。使用该模型，可以很轻松的创建自适应浏览器窗口的流动布局或自适应字体大小的弹性布局。

1. 网页编码UTF-8、GBK和GB2312编码的区别？（重要）

GBK通常指GB2312编码 只支持简体中文字

utf通常指UTF-8，支持简体中文字、繁体中文字、英文、日文、韩文等语言（支持文字更广）

通常国内使用utf-8和gb2312，看自己需求选择

1. html文档声明区分大写吗？（重要）

doctype 声明是不区分大小写的，以下方式均可：

<!DOCTYPE html>、<!DOCTYPE HTML>、<!doctype html>和<!Doctype Html>

1. 如何实现关闭表单内所有控件的自动完成功能？（重要）

<form autocomplete=”off”></form>

1. 刷新页面表单内第一个控件自动定位光标，如何实现？

<form>

<input type=”text”autofocus>

</form>

1. 有一个表单，里面有上传文件框，请问这个表单应该设置那些属性？

<form method=”POST” enctype=”mutipart/form-data”></form>

1. 单选框、复选框和下拉列表框默认选中属性是什么？（重要）

单选框和复选框是checked，下拉列表框是selected

1. display有那些值？

内联：inline

块级：block

行块：inline-block

表格：table

行: table-row

单元格: table-cell

列表：list-item

隐藏：none

1. 表格单元格合并列、合并行、强制不换行分别是什么？（重要）

合并列：colspan;

合并行：rowspan

强制不换行：nowrap

1. 项目列表有那些？

有序列表：ul

无序列表：ol

自定义列表：dl

1. form表单标签有什么特点？表单属性分别表示什么？

表单特点：  
收集用户填写的信息提交给后台处理；

表单标签是块级标签

一个页面可以有很多个表单标签，但不可以使用嵌套  
表单属性：  
name：定义表单名称；

action：提交后台处理文件的地址

methd：表单提交的方式（get/post）

target：提交的目标窗口

enctype：设置请求文档的编码类型

1. get和post的区别是？（必须）

1. GET是不安全的，因为在传输过程，数据被放在请求的URL中，而POST的所有操作对用户来说  
都是不可见的。

2. GET传送的数据量较小，这主要是因为受URL长度限制，而POST传送的数据量较大，一般被默  
认为不受限制。

3. GET限制表单数据集的值必须为ASCII字符，而POST支持整个ISO10646字符集。

4. GET执行效率却比POST方法好，GET是表单提交的默认方法。

1. 如何在地址栏出现小图标？（重要）

<head>  
<link rel="shortcut icon" href="favicon.ico"/>  
</head>

1. pre标签有什么用？（重要）

pre元素可定义预格式化的文本。被包围在pre元素中的文本通常会保留空格和换行符。而文本也会呈现为等宽字体。

<pre>标签的一个常见应用就是用来表示计算机的源代码。

1. 简述一下你对HTML语义化的理解？（必须）

用正确的标签做正确的事情。

结构更清晰，html语义化让页面的内容结构化，便于对浏览器、搜索引擎解析。  
方便SEO优化，搜索引擎的爬虫也依赖于HTML标记来确定上下文和各个关键字的权重。

便于阅读维护理解，使阅读源代码的人对网站更容易将网站分块。

1. label的作用是什么？是怎么用的？（重要）

label通过包裹元素或者使用for属性的方式扩大控件触发的范围

可以使用for属性指向元素id值或者用label标签包裹元素来实现扩大触发范围

1. iframe有那些优缺点？（必须）

优点：  
a. 页面中可以嵌入其他页面

b. 可以实现局部刷新

c. 模块分离，便于更改，如果有多个网页引用iframe，只需要修改iframe的内容，就可以实现调用  
的每一个页面内容的更改，方便快捷

d. 重载页面时不需要重载整个页面，只需要重载页面中的一个框架页

缺点：

a. 因为iframe中src指向一个独立页面，所以样式和脚本需要额外链入，增加http请求。

b. 代码结构变复杂了，也不利于SEO优化。

c. 框架结构有时会让人感到迷惑，比如出现双滚动条，影响用户体验。

d. 产生多个页面，不易管理

e. 移动端兼容性不是特别好

f. 阻塞页面渲染，性能差。

1. 实现不使用 border 画出1px高的线，在不同浏览器的标准模式与怪异模式下都能保持一致的效果。

<div style="height:1px;background:red;overflow:hidden;"></div>

1. title与h1的区别、b与strong的区别、i与em的区别？（必须）

title属性设置窗口标题，h1则表示层次明确的标题，对页面信息的抓取也有很大的影响；

strong是标明重点内容，有语气加强的含义，使用阅读设备阅读时<strong>会重读,，而<b>是展示强调内容。

i内容展示为斜体，em表示强调的文本；

1. 请描述下SEO中的TDK？（必须）

在SEO中，所谓的TDK其实就是title、description和keywords这三个标签，这三个标签在网站的优化过程中title标题标签，description描述标签，keywords关键词标签

1. 标签上title与alt属性的区别是什么？（必须）

alt是给搜索引擎识别，在图像无法显示时的替代文本

title是关于元素的注释信息，主要是给用户解读

1. 前端页面有哪三层构成，分别是什么？作用是什么？（重要）

分成：结构层HTML、表示层CSS、行为层Javascript。

1. 每个HTML文件头里都有个很重要的东西，Doctype，知道这是干什么的么？

<!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前。

作用

1. 告知浏览器文档使用哪种 HTML 或 XHTML 规范。
2. 告诉浏览器按照何种规范解析页（如果你的页面没有DOCTYPE的声明，那么compatMode默  
    认就是BackCompat,浏览器按照自己的方式解析渲染页面）
3. DIV+CSS布局较table有什么优势？（重要）
4. 速率更快，页面体积变小，浏览速度变快，这就使得对于某些控制主机流量的网站来说是最大的  
    优势了
5. 更好地被搜索引擎收录，大部分html页面的样式写入了CSS文件中，便于被搜索引擎采集收录；
6. 对浏览者和浏览器更具优势，由于CSS富含丰富的样式，使页面更加灵活性，它可以根据不同的  
    浏览器，而达到显示效果的统一和不变形；
7. 修改更有效率，由于使用了DIV+CSS制作方法，在修改页面的时候更加容易省时，提高工作效  
    率；
8. 为什么用多个域名存储网站资源更有效？（重要）

1、CDN缓存更方便

2、突破浏览器并发限制

3、节约cookie带宽

4、节约主域名的连接数，优化页面响应速度

5、防止不必要的安全问题

1. 简要说明一下做好SEO需要考虑哪些方面？（重要）

1、TDK

2、图片+alt

3、友情链接

4、站点地图

1. 上下margin会存在重合的问题，如何解决？（重要）

在重合元素外包裹一层容器，并触发该容器生成一个BFC（overflow：hidden）

1. 表格涉及那些标签？

table 表格 tr 行 th 字段 td 单元格 caption 标题 colgroup 列组 col 列 thead表格头  
tbody 表格主体 tfoot表格脚部

1. 图像src属性支持几种方式的图片导入？

支持本地图片、网络图片、base64内嵌图片和blob协议图片

1. alt的主要用途是什么？（重要）

当图片加载不出来时alt属性中的内容会显示出来，搜索引擎会根据alt设置文字内容收录图片信息

1. img的alt与title有何异同？ strong与em的异同？（必须）

alt属性：为不能显示图像的用户代理，alt属性用来指定替换文字。  
替换文字的语言由lang属性指定。(在IE浏览器下会在没有title时把alt当成 tool tip显示)

title属性：该属性为设置该属性的元素提供建议性的信息。

em：表现为斜体，语义表示强调

strong：表现为粗体，语义为强烈语气，强调程度超过em

1. 简述一下src与href的区别？（重要）

src用于替换当前元素

href用于在当前文档和引用资源之间确立联系

src是source的缩写，指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置

href是Hypertext Reference的缩写，指向网络资源所在位置，建立和当前元素（锚点）或当前

1. 如何实现页面每隔5秒自动刷新或者5秒后自动跳转到百度搜索页面？

<meta http-equiv=”refresh” content=”5;Url=https://www.baidu.com”/>

1. 什么是base64编码图片？它的优缺点是什么？使用建议是？

将图片转码为字符串形式代替图片地址进行加载，内嵌再html文档中

优点:减少请求次数，减轻服务器压力

缺点:编码后体积会变大，文件请求速度会变慢

建议:一般使用8-12kb大小，颜色较为单一的图片

1. 行块元素之间存在间隙，是什么造成的？如何解决？（必须）

间隙是元素之间存在空格造成的，解决方法有两种：  
1. 将父容器设置font-size：0，这样可以将空格间距去掉，兼容性一般，代码易读

1. 将元素首尾紧密相连不要有任何间隙，兼容性最好，代码不易读
2. mark和bdo标签是什么？

mark是html5新增的文本高亮标签

bdo是设文本排列方向的标签，默认从左向右，也可以从右向左

1. html5新增那些语义化标签？（必须）

header 头部 footer脚部 main主体 section区块 aside侧边栏 article文章  
details 详细 figure 图文 nav导航 menu菜单

1. 简述HTML5新增的canvas、audio、svg标签的作用？（重要）

canvas被称作画布，canvas 元素使用 JavaScript 在网页上绘制图像。   
画布是一个矩形区域,可以控制其每一像素。

audio标签可以引用音频资源，在页面上播放音乐

svg用XML格式定义图形，可以用来制作矢量图形。

1. 如何在 HTML5 页面中嵌入音频？

HTML 5 包含嵌入音频文件的标准方式，支持的格式包括 MP3、Wav 和 Ogg：

<audio controls>

<source src="jamshed.mp3" type="audio/mpeg">

Your browser does'nt support audio embedding feature.

</audio>

1. 如何在HTML5页面中嵌入视频？

和音频一样，HTML5 定义了嵌入视频的标准方法，支持的格式包括：MP4、WebM 和 Ogg：

<video width="450" height="340" controls>

<source src="jamshed.mp4" type="video/mp4">

Your browser does'nt support video embedding feature.

</video>

1. 网页颜色定义有几种形式?

1）十六进制 例如：#FF0000表示红色

2）rgb 释义：r表示红色 g表示绿色 b表示蓝色

3）rgba 释义：a表示alpha通道透明

4）颜色名 例如：red表示红色

5）hsl 释义：h表示色相、s表示饱和度、l表示亮度

1. 16进制颜色是怎么缩写的?

#FF0000表示红色 可以缩写成 #F00

#00FF00表示绿色 可以缩写成 #0F0

#000000表示黑色 可以缩写成 #000

1. 使用rgba定义一个半透明的红色？

<div style=”color:rgba(255,0,0,0.5)”></div>

1. jpg、png、gif、svg和webp图片格式区别？（必须）

jpg：有损压缩格式图片，是位图，通常用于产品、风景、人物等色彩丰富的图片。

png：无损格式图片，图片画质清晰，支持alpha通道透明，通常用于小图标、或者产品图片。

gif：以8位色重现真色彩的图像，所以一般用于图标，或者颜色单一的图片，其次它可以做动图。

svg：svg是矢量图片，支持js事件，同事支持图片动画。

webp：这是谷歌发明的一款优于jpg的图片压缩格式，压缩率只有jpg的2/3。

1. 知道的网页制作会用到的图片格式有哪些？（重要）

png-8，png-24，jpeg，gif，svg。

但是上面的那些都不是面试官想要的最后答案。面试官希望听到是Webp。（是否有关注新技术，新鲜事物）科普一下Webp：WebP格式，谷歌（google）开发的一种旨在加快图片加载速度的图片格式。图片压缩体积大约只有JPEG的2/3，并能节省大量的服务器带宽资源和数据空间。Facebook Ebay等知名网站已经开始测试并使用WebP格式。

在质量相同的情况下，WebP格式图像的体积要比JPEG格式图像小40%

1. HTML5 为什么只需要写 <!DOCTYPE HTML>？（重要）

HTML5 不基于 SGML（标准通用标记语言），因此不需要对DTD进行引用，但是需要doctype来规范浏览器的行为（让浏览器按照它们应该的方式来运行）。

HTML4.01基于SGML,所以需要对DTD进行引用，才能告知浏览器文档所使用的文档类型。

1. 表单提交给后端，后端是根据什么属性来获取表单控件的值？（重要）

根据name获取控件对象，然后根据 value获取用户的数据。

大前端 弱化了form标签的功能，可以做到少用、甚至不用。

1. video如何实现自动播放功能？

chrome浏览器为了避免用户一打开视频就听到视频播放声音而受到惊吓。为了提升用户体验，因此默认禁止了自动播放的功能，除非你设置为静音模式。

1. 如何实现 单击超级链接，再iframe中显示？

首先要有一个超级链接，给它的属性target取一个名字，其次要有一个iframe，给它取名main

<a href="about.html" target="main">关于我们</a>

<iframe src="home.html" name="main" width="200" height="200"></iframe>

1. 请介绍一下head标签下有那些标签，具体有什么作用？

<meta charset="utf-8"/> 网站编码

<meta name="viewport"/> 视口，控制页面缩放比例

<meta name="keywords"/> 关键字，方便`SEO`优化

<meta name="description"/> 描述，方便`SEO`优化

<meta name="author"/> 作者

<title>标题</title> 标题

<link> 外部样式

<style> 内部样式

<base> 链接基础地址

<script> js脚本

1. <input type="file"> 有那些属性？分别是什么？

multiple 允许上传多个文件

accept 允许上传文件类型

webkitdirectory 允许上传文件夹

value 文件物理伪装路径

1. 如何使用超级链接打开邮箱或打电话？（重要）

<!--打开邮箱-->  
<a href="mailto:xianweb@qq.com" targer="\_blank">发送邮件</a>  
  
<!--拨打电话-->  
<a href="tel:18579105168" targer="\_blank">打给我</a>

1. canvas 和 svg有什么区别？（必须）

canvas是通过JavaScript代码来绘制2d图形的

svg是通过XML来描述2d图形的

svg是矢量图，放大不失真，而canvas是位图，放大失真出现锯齿效果

svg里面的每个节点都可以单独添加事件，而canvas只可以给画布添加事件

svg渲染速度慢；而canvas渲染速度快，适合游戏场景制作

CSS专题

1. CSS盒子模型有几种？分别是什么？（必须）

有两种， IE 盒子模型（ border-box ）、W3C 盒子模型（ content-box ）

盒模型： 内容(content)、填充(padding)、边界(margin)、 边框(border)；

区别： IE的content部分把 border 和 padding计算了进去。

默认是 content-box 内容盒子。

1. src 和 href的区别是？（必须）

src用于替换当前元素，href用于在当前文档和引用资源之间确立联系。

1. a超级链接有几个状态，它的顺序是？（必须）

顺序按照 L-V-H-A 排序

1. a:link : 默认状态

2. a:visited: 单击过后的状态

3. a:hover：划过时的状态

4. a:active：单击一瞬间的状态

1. css样式引入有几种方式？（重要）

外部引入（项目文件比较多推荐这种方式，但增加http请求）

内部引入样式

行内引入样式

1. :target伪类有什么用？请举例说明

:target是锚点伪类选择器,用来改变页面中锚链接URL所指向的ID元素的样式

:target{  
 color: red;  
}  
<a href="#home">首页</a>  
<a href="#about">关于</a>  
<div id="home">首页</div>  
<div id="about">关于</div>

1. 伪类和伪元素的区别？（重要）
2. 伪类使用单冒号
3. 伪元素使用双冒号，有时为了兼容性也可以使用单冒号
4. 伪元素会创造一些不存在原文档结构的元素，它也不可复制，也没有具体的标签名字
5. 伪类表示已经存在的某个元素处于某种状态，这种状态dom树无法表示，这时候就可以通过伪类  
    来弥补这里的不足
6. 伪类和伪元素同时作用的话如何排序？举例说明

同时用的时候伪类必须放在伪元素的前面使用，像以下这样，因为伪元素不是实体元素

.box:hover::after{content: "\20";}

1. css选择器的优先级和权重分别是？（必须）

!important > 行内样式 > id样式 > （class = 属性 = 伪类） > 标签 > 默认  
按大小顺序是：

a. !important 权重为无穷大，!important会带来性能开销

b. style行内样式，权重为1000

c. Id选择器，权重为100

d. 类选择器=属性选择器=伪类选择器，权重为10

e. 伪元素选择器=标签选择器，权重为1

f. 通配符选择器，权重为0

h. 默认样式，权重为0

1. CSS3新增伪类有那些？

p:first-of-type 选择属于其父元素的首个元素

p:last-of-type 选择属于其父元素的最后元素

p:only-of-type 选择属于其父元素唯一的元素

p:only-child 选择属于其父元素的唯一子元素

p:nth-child(2) 选择属于其父元素的第二个子元素

:enabled :disabled 表单控件的禁用状态。

:checked 单选框或复选框被选中。

1. 文本属性有那些？（重要）

1. color 字体颜色

1. font-size 字体大小
2. font-family 字体名称

4 .font-wight 字体加粗

1. font-style 字体倾斜
2. font-variant 大写字母
3. text-align 文本水平对齐
4. text-transform 文本变形
5. text-shadow 文本阴影
6. text-indent 文本缩进
7. text-overflow 文本溢出

12 .text-decoration 文本修饰线

13 .line-height 行高

1. word-wrap 单词换行
2. white-space 强制不换行
3. vertical-align 垂直对齐
4. 设置元素浮动后，该元素的display值是多少？（必须）

会自动变成块级 display:block。

1. 使用 CSS 预处理器吗？

Less 和 Sass

1. 简述一下Sass、Less，且说明区别？（重要）

他们是动态的样式语言，是CSS预处理器,CSS上的一种抽象层。他们是一种特殊的语法/语言而编译成CSS。

区别：

变量符不一样，less是@，而Sass是$;

Sass支持条件语句，可以使用if{}else{},for{}循环等等。而Less不支持;

Sass是基于Ruby的，是在服务端处理的，而Less是需要引入less.js来处理Less代码输出Css到浏览器

1. 满足那些条件之一就可触发BFC？（重要）

1. 根元素，即html

2. float的值不为none（默认）

3. overflow的值不为visible（默认）

4. display的值为inline-block、table、table-cell、table-caption等

5. position的值为absolute或fixed

1. \*height 、\_height 和 +height 的区别？

\*height: 100px /\*IE7及以下能识别\*/  
\_height: 100px /\*IE6能识别\*/  
+height: 100px /\*IE7能识别\*/

1. 为什么要初始化CSS样式？（重要）

浏览器差异

因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。

提高编码质量

初始化CSS为我们节约网页代码，节约网页下载时间；还会使得我们开发网页内容时更加方便简洁，不用考虑很多。

1. inline-block默认的对其方式是什么？在使用inline-block时在内容不同的时候想要保持内容水平顶对齐，说一下你采用的方法？

默认对齐方式：baseline 基线对齐

内容水平顶对齐方案：vertical-align:top; 顶对齐

1. 行(inline-block)块元素之间存在间隙?是什么导致的？如何解决？（必须）

是行块元素之间的空白符造成的，解决方法有以下几种：

行块元素紧密相连

父元素设置font-size: 0

行块元素使用浮动

1. 表格边框默认分明边和暗边，如果合并这两条边？（重要）

在css样式中设置为收缩性边框

table{border-collapse: collapse;}

1. 对于这种1111111aaaaa连续不间断文本，如何让它强制换行？（重要）

.box{

word-break: break-all;

/\*或者\*/

word-break: keep-all;

}

break-all：允许单词间换行，可能导致单词不完整  
keep-all：允许空格间换行，单词不会截断。

1. CSS 都有哪些选择器？（必须）
2. 标签选择器（ 如：body,div,p,ul,li ）
3. 类选择器（如：class="head", class="head\_logo" ）
4. ID选择器（如：id="name", id="name\_txt" ）
5. 通配符选择器（如：\*号）
6. 组合选择器（如：.head .head\_logo, 注意两选择器用空格键分开）
7. 继承选择器（如：div p, 注意两选择器用空格键分开）
8. 伪类选择器（如：链接样式, a元素的伪类，4种不同的状态：link、visited、active、hover。）
9. px、em、rem 单位之间的区别？（必须）

px 相对长度单位，像素px是相对于显示器屏幕分辨率而言的。   
em 相对长度单位，会继承父级元素的字体大小，若父元素字体大小为14px，。  
rem 是相对于根元素html字体大小的单位。

1. display:none和visibility:hidden的区别？（必须）

display:none和visibility:hidden都可以用于元素的隐藏

两者之间的区别：

display:none隐藏不占据文档流的位置。

visibility:hidden会占据文档流中的位置。

1. 简明说一下link和@import的区别和用法？（重要）

1. link是XHTML标签，除了加载CSS外，还可以定义RSS等其他事务，@import属于CSS范畴，  
 只能加载CSS。

2. link引用CSS时，在页面载入时同时加载，@import需要页面网页完全载入以后加载。

3. link是XHTML标签，无兼容问题，@import是在CSS2.1提出的，低版本的浏览器不支持。

4. link支持使用js控制DOM去改变样式，而@import不支持。

1. CSS背景的缩写规则是？

background: color 颜色 | image 图像 | repeat 重复 | attachment 附件 | position 定位 / size 尺寸

1. 你对line-height是如何理解的？

行高是指一行文字的高度，具体说是两行文字间基线的距离。CSS中起高度作用的是height和line-height，没有定义height属性，最终其表现作用一定是line-height。

单行文本垂直居中：把line-height值设置为height一样大小的值可以实现单行文字的垂直居中，其实也可以把height删除。

多行文本垂直居中：需要设置display属性为inline-block。

1. CSS文本格式化字体修饰font这块如何缩写？

font: font-style 倾斜 | font-variant 大小写 | font-weight 加粗 | font-size 尺寸 / line-height 行高 | font-family 字体

1. 字体类型有几种？请举几个有代表性的字体

1. 衬线字体 Times New RomanGeorgia 通常用于正文显示

2. 非衬线字体 Arial Verdana 标题

3. 等宽字体 Courier New Lucida Console 源代码书写

1. 单行和多行溢出省略如何实现？（必须）

单行溢出省略

overflow: hidden;

width: 100px;

white-space: nowrap;

text-overflow: ellipsis;

多行溢出省略

overflow: hidden;

width: 100px;

display: -webkit-box;

-webkit-line-clamp: 3;

-webkit-box-orient: vertical;

text-overflow: ellipsis;

1. CSS盒子模型有几种？分别是什么？（必须）

有两种， IE 盒子模型（ border-box ）、W3C 盒子模型（ content-box ）

盒模型： 内容(content)、填充(padding)、边界(margin)、 边框(border)；

区别： IE的content部分把 border 和 padding计算了进去。

默认是 content-box 内容盒子。

1. 请用纯CSS写一个▲三角形？（重要）

<style>

.arrow{

width: 0;

height: 0;

border: 10px solid transparent;

border-top: none;

border-bottom-color: black;

}

</style>

<span class="arrow"></span>

1. transition和animation的区别？（重要）

1. transition 需要触发事件，比如点击事件、鼠标移入事件；而 animation可以配合关键帧 可以  
 不触发事件就触发这个动画

2. transition 触发一次播放一次；而 animation 则是可以设置很多的属性，比如循环次数，动画结  
束的状态等等；

3. transition关注的是样式属性的变化，而animation作用于元素本身使用关键帧的概念，实现更  
自由的动画效果；

4. transition动画比animation性能开销要低，不会引起页面重排。

1. relative 相对定位、fixed固定定位和absolute的区别是什么？（必须）

1. 相对定位 relative：不脱离文档流，参考自身静态位置通过top，bottom，left，right定位，并  
且可通过z-index进行层次分级。

2. 绝对定位 absolute：脱离文档流，通过top，bottom，left，right定位，选取其最近的父级元  
素进行定位，当父级元素position为static时，absolute元素将以body坐标原点进行定位，可通过z-index层次分级。

3. 固定定位 fixed：这里他所固定的对象是可视窗口而并非body，或父级元素，可通过z-index进  
行层次分级。

1. 如果需要手动写动画，你认为最小时间间隔是多久，为什么？（重要）

多数显示器默认频率是60Hz，即1秒刷新60次，所以理论上最小间隔为1/60＊1000ms ＝ 16.7ms

1. CSS属性overflow属性定义溢出元素内容区的内容会如何处理？

参数是scroll时候，必会出现滚动条。

参数是auto时候，子元素内容大于父元素时出现滚动条。

参数是visible时候，溢出的内容出现在父元素之外。

参数是hidden时候，溢出隐藏。

1. 清除浮动有几种办法？（必须）

利用创建bfc清除，在浮动元素 overflow:hidden，position：absolute等  
在父元素内部最后面，添加额外块级元素，并设置clear:both  
使用::after伪元素避免添加额外的标签

.clearfix::after{

display: block;

content: " ";

clear: both;

visibility: hidden;

line-height: 0;

}

使用display: table;触发BFC块格式化上下文, 限制其内部元素float的不影响到BFC外面。

.clearfix:before,

.clearfix:after {

display: table;

line-height: 0;

content: "";

}

.clearfix:after {

clear: both;

}

1. 设置那些CSS属性，会让元素自动转换为块级block特性？（重要）

设置定位 absolute 和 fixed

设置浮动 float

设置弹性布局 display: flex

设置表单 display: table

设置单元格 display: table-ceil

1. 有那些样式会使得元素脱离文档流？（重要）

浮动 float

定位（ absolute、fixed、sticky ）

display:none

1. IE8-(IE8及以下)rgba模式不兼容的解决方案？

IE8以及以下用滤镜

.opacity{

filter: alpha(opacity=20);

}

1. CSS3有哪些新特性？（必须）

新增伪类伪元素选择器

弹性盒模型 display: flex;

媒体查询

个性化字体

颜色透明度

圆角 渐变 阴影 文字溢出 背景效果 边框效果 变形

动画（过渡和自定义动画）

1. 请解释一下CSS3的flex（弹性盒布局模型），以及适用场景？

布局的传统解决方案，基于盒状模型，依赖 display 属性 + position属性 + float属性。它对于那些特殊布局非常不方便，比如，垂直居中就不容易实现，此外实现各类对齐（两端对齐、均分对齐）也很麻烦。  
Flex 布局，可以简便、完整、响应式地实现各种页面布局。

Flex 布局分主轴（水平）、交叉轴（垂直）两种轴向。  
  
Flex 容器的属性分：  
1. flex-direction 排列方向（水平还是垂直）

2. flex-wrap 强制换行，默认强制不换行

3. flex-flow 属性是 flex-direction 和 flex-wrap 属性的复合属性。

4. justify-content 水平对齐方式（左对齐、右对齐、居中对齐、两端对齐、均分对齐）

5. align-items 垂直对齐方式（顶对齐 居中对齐 低对齐 基线对齐 拉伸对齐）

6. align-content 定义了多根轴线的对齐方式，也就是多行的情况  
  
项目的属性（子元素上的属性）：

1. order 排序，数字越小越靠左

2. flex-grow 项目的放大比例, 默认为0，即如果存在剩余空间，也不放大。

3. flex-shrink 项目的缩小比例，默认为1，即如果空间不足，该项目将缩小。

4. flex-basis 类似于于width，但优先级高于width属性。

5. align-self 属性允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。

1. 弹性盒子模型? flex|box区别?（重要）
2. 引入弹性盒布局模型的目的是提供一种更加有效的方式来对一个容器中的条目进行排列、对齐和分配空白空间。即便容器中条目的尺寸未知或是动态变化的，弹性盒布局模型也能正常的工作。在该布局模型中，容器会根据布局的需要，调整其中包含的条目的尺寸和顺序来最好地填充所有可用的空间。

当容器的尺寸由于屏幕大小或窗口尺寸发生变化时，其中包含的条目也会被动态地调整。比如当容器尺寸变大时，其中包含的条目会被拉伸以占满多余的空白空间；当容器尺寸变小时，条目会被缩小以防止超出容器的范围。弹性盒布局是与方向无关的。

在传统的布局方式中，block 布局是把块在垂直方向从上到下依次排列的；而 inline 布局则是在水平方向来排列。弹性盒布局并没有这样内在的方向限制，可以由开发人员自由操作。

1. flex和box的区别: display：box 是老规范，要兼顾古董机子就加上它； 父级元素有display:box;属性之后。他的子元素里面加上box-flex属性。可以让子元素按照父元素的宽度进行一定比例的分占空间。flex是最新的，董机老机子不支持的；

父元素设置display:flex后，子元素宽度会随父元素宽度的改变而改变，而display:box不会。 Android UC浏览器只支持display: box语法；而iOS UC浏览器则支持两种方式。

1. 移动端的布局用过媒体查询吗？（重要）

通过媒体查询可以为不同大小和尺寸的媒体定义不同的CSS，适应相应的设备的显示。

媒体查询和头部视口设置有关系，通常设置不缩放

<head>  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
</head>

比如文档宽度小于 300 像素则修改背景颜色

@media screen and (max-width: 300px) {  
 body {  
 background-color: red;  
 }  
}

1. 什么是响应式设计？响应式设计的基本原理是什么？（重要）

响应式网站设计(Responsive Web design)是一个网站能够兼容多个终端，而不是为每一个终端做一个特定的版本。

基本原理是通过媒体查询(@media screen)检测不同的设备屏幕尺寸做处理。

页面头部必须有meta声明的viewport。

<head>  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
</head>

1. CSS优化、提高性能的方法有哪些？

避免过度约束

避免后代选择符

避免链式选择符

使用紧凑的语法

避免不必要的命名空间

避免不必要的重复

最好使用表示语义的名字。一个好的类名应该是描述他是什么而不是像什么

避免!important，可以选择其他选择器

尽可能的精简规则，你可以合并不同类里的重复规则

1. ::before 和 :after中双冒号和单冒号有什么区别？（重要）

单冒号(:) 用于CSS3伪类

双冒号(::)用于CSS3伪元素。

::before就是以一个子元素的存在，定义在元素主体内容之前的一个伪元素。并不存在于dom之中，只存在在页面之中。

:before 和 :after 这两个伪元素，是在CSS2.1里新出现的。起初，伪元素的前缀使用的是单冒号语法，但随着Web的进化，在CSS3的规范里，伪元素的语法被修改成使用双冒号，成为::before ::after

1. 让页面里的字体变清晰，变细用CSS怎么做？(了解)

-webkit-font-smoothing在window系统下没有起作用，但是在IOS设备上起作用-webkit-font-smoothing：antialiased是最佳的，灰度平滑。

1. style标签写在body后与body前有什么区别？(了解)

页面加载自上而下 当然是先加载样式。

写在body标签后由于浏览器以逐行方式对HTML文档进行解析，当解析到写在尾部的样式表（外联或写在style标签）会导致浏览器停止之前的渲染，等待加载且解析样式表完成之后重新渲染，在windows的IE可能会出现FOUC现象（即样式失效导致的页面闪烁问题）

1. 阐述一下CSS Sprites（雪碧图）？（重要）

将一个页面涉及到的所有图片都包含到一张大图中去，然后利用CSS的 background-image，background- repeat，background-position的组合进行背景定位。利用CSS Sprites能很好地减少网页的http请求，从而大大的提高页面的性能；CSS Sprites能减少图片的字节。

1. 如何解决小于12px的字体大小？（重要）

1. 用图片替换文字

2. 使用css中transform：scale

3. svg 字体填充

1. font-family属性注意什么？

1. 设置的字体要考虑用户的设备上有没有这种字体

2. 设置体字尽量用英文字体，最好不要用中文字

3. 如果是组合的单词，需要加双引号

4. 字体支持后备机制，使用多种字体，前面的字体没有，就使用后的字体

5. 可以使用webfont技术定制字体保证兼容性

1. CSS如何实现强制不换行？

white-space: nowrap;

1. CSS实现强制换行（重要）

div{  
 word-wrap: break-word;  
 word-break: normal;  
}

强制英文单词断行

div{  
 word-break:break-all;  
}

1. rgba 和 opacity 的区别？（重要）

子元素opacity会继承父元素的 opacity 属性，后代元素的颜色都变透明。  
RGBA设置的元素的后代元素不会继承不透明属性，只让当前元素的颜色变透明。

1. 段落首行缩进2个字如何实现？

.text{  
 text-indent: 2em;  
}

1. 行高的倍数是如何计算的？公式是什么？

行高的倍数 = 要达到的行高像素/字体大小

1. word-break 单词换行方式有那些？

break-all 不考虑单词完整性换行

keep-all 保持单词完整性换行

1. background-image: url(图片地址); 中的图片地址要不要加引号？

推荐不要，因为部分浏览器加了引号不支持

1. background-origin 和background-clip的区别？

它们的参数是 content-box、padding-box、border-box

background-origin定义的是背景位置填充起始点，默认是content-box

而background-clip是对背景的切割裁切位置，默认是border-box

1. color属性中 initial 和 inherit 有什么区别？

inhert会继承父类元素的颜色

initial会保持最初的颜色

1. 能否通过css代码调整背景图片的尺寸？如何实现？

能，background-size: 100% 100%;

1. 盒子模型包含那些属性？ （重要）

width宽、height高、padding填充、border边框、margin边距

1. 光标有那些类型？

default 默认

pointer 手型

move 移动

not-allowed 禁止

s-resize 垂直调整

w-resize 水平调整

1. iconfont字体使用步骤 （重要）

登录iconfont.cn网站

在iconfont.cn中创建我的图标项目

将自己画的图标导入iconfont.cn项目中

在项目中使用iconfont图标（1. unicode编码 2. font class 字体类 3. symbol 符号）

1. 如何在项目中使用webfont字体？ （重要）
2. 首先要有字体文件
3. 在自己的项目中创建fonts文件夹，将字体文件拷贝到其内
4. @font-face定义web字体
5. 使用定义的web字体
6. 引用web字体
7. 伪类选和伪元素选择器分别有那些？

伪类：  
:first-child 第一个

:not 匹配除了谁（不包含于）

:target 锚点

:hover 划过

:nth-child(n) 根据索引值选择标签

:checked 匹配选中标签（单选框和复选框）

伪元素：  
::before 在元素内部的前面创建

::after 在元素内部不的后面创建

::first-letter 首字母

::first-line 首行第一行

::selection 选区

::placeholder 占位提示文字

webkit 滚动条

1. CSS选择符有哪些？哪些属性可以继承？（必须）

1、id选择器（ # myid）

2、类选择器（.myclassname）

3、标签选择器（div, h1, p）

4、相邻选择器（h1 + p）

5、子选择器（ul > li）

6、后代选择器（li a）

7、通配符选择器（ \* ）

8、属性选择器（a[rel = "external"]）

9、伪类选择器（a:hover, li:nth-child）

可继承的样式： font-size font-family color, UL LI DL DD DT;

不可继承的样式：border padding margin width height

1. 低版本浏览器不支持display:inline-block属性，请问怎么兼容？

.box{

display: inline-block;

\*display:inline;

\*zoom:1;

}

1. margin-left:calc(-100%-100px) 代码中用到了一个calc（），这个函数的作用是什么？

通过计算来确定CSS属性值。

calc是英文单词calculate(计算)的缩写，是css3的一个新增的功能，你可以使用calc()给元素的border、margin、pading、font-size和width等属性设置动态值。calc()可以使用数学运算中的简单加（+）、减（-）、乘（\*）和除（/）来解决问题，而且还可以根据单位如px,em,rem和百分比来转化计算

1. CSS的单位中，设定元素的长度或宽度与父元素字体大小相关的单位是什么？  
    与html文档元素大小相关的单位是什么？

单位分别是em和rem

1. px、em、rem、vh、vw的区别是什么？（必须）

px代表物理屏幕上能显示出的最小的一个点,

em 是相对于父级的字体大小,

rem是相对于HTML根元素的字体大小,

vh 和vw相对于视口的高度和宽度,1vh 等于1/100的视口高度，1vw 等于1/100的视口宽度

1. FontAwesome和iconfont是什么？他们有什么异同，问什么要使用它，有什么弊端？（重要）

两个都是图标字体，Font Awesome 是一套完美的图标字体

Iconfont是阿里妈妈的在线字体库平台，可以定制自己要的字体图标。

图标字体的优势：

1、轻量性：一个图标字体比一系列的图像（特别是在Retina屏中使用双倍图像）要小。一旦图标字  
体加载了，图标就会马上渲染出来，不需要下载一个图像。可以减少HTTP请求，还可以配合HTML5离线存储做性能优化。

2、灵活性：图标字体可以用过font-size属性设置其任何大小，还可以加各种文字效果，包括颜色、  
Hover状态、透明度、阴影和翻转等效果。可以在任何背景下显示。使用位图的话，必须得为每个不同大小和不同效果的图像输出一个不同文件。

3、兼容性：网页字体支持所有现代浏览器，包括IE低版本。详细兼容性可以点击这里。

图标字体的缺点：

1、只能单色

2、跨域问题

3、字体图标库似乎体积显得有些过大

1. 解释下这个CSS选择器什么发生什么？  
    [role=navigation] > ul a:not([href^=mailto]) {}

定义了role属性，并且值为navigation的任何元素，其子元素列表下的除邮箱链接之外的所有链接元素。

1. 网页前端性能优化的方式有哪些？（必须）

1. 压缩css, js, 图片

2. 减少http请求次数，合并css,js 合并图片（雪碧图）

3. 使用CDN

4. 减少dom元素数量

5. 图片懒加载

6. 静态资源另外用无cookie的域名

7. 减少dom的访问（缓存dom）

8. 巧用事件委托  
9. 样式表置顶、脚本置低

70. 移动端兼容性？（重要）

1、ios移动端click事件300ms的延迟响应:  
 用fastclick来解决  
 zepto的touch模块的tap事件有点透的问题，也是用fastclick来解决

2、一些情况下对非可点击元素如(label,span)监听click事件，ios下不会触发，css增加  
cursor:pointer就搞定了

3、ios下取消input在输入的时候英文首字母的默认大写

<input autocapitalize="off" autocorrect="off" />

4、fixed定位缺陷：  
ios下fixed元素容易定位出错，软键盘弹出时，影响fixed元素定位  
android下fixed表现要比iOS更好，软键盘弹出时，不会影响fixed元素定位  
ios4下不支持position:fixed，可用iScroll插件解决这个问题

5、Input 的placeholder会出现文本位置偏上的情况：  
input 的placeholder会出现文本位置偏上的情况：PC端设置line-height等于height 能够对齐，而移动端仍然是偏上，解决是设置line-height：normal

6、calc的兼容性处理：  
CSS3中的calc变量在iOS6浏览器中必须加-webkit-前缀，目前的FF浏览器已经无  
-moz-前缀。  
Android浏览器目前仍然不支持calc，所以要在之前增加一个保守尺寸：

div {

width: 95%;

width: -webkit-calc(100% - 50px);

width: calc(100% - 50px);

}

7、在移动端修改难看的点击的高亮效果，iOS和安卓下都有效：

\* {-webkit-tap-highlight-color:rgba(0,0,0,0);}

8、阻止旋转屏幕时自动调整字体大小

html, body, form, fieldset, p, div, h1, h2, h3, h4, h5, h6 {-webkit-text-size-adjust:none;}

71. 谈谈你对bootstrap的认识，用它做过项目么？(了解)

用Bootstrap主要用来做一些响应式的网站，它里面栅格化这块比较强大，我可以针对不同的屏幕尺寸定制不同页面结构。

我还用它做过后台管理系统，因为它里面集成的有风格标准统一的组件、插件、全局样式等，是我们做后台管理系统的话，不用再花大精力去布局页面了。而且甚至产品给我们一个原型图我们就可以开始开发，大大提高了开发效率。

1. 如果已经开发完一个web应用用于手机端访问，请问需要在哪方面优化页面？  
   （重要）

1. 按需加载

1. 静态资源压缩
2. 巧用图标字体
3. 保证首屏相应速度
4. 动画能用css3实现的尽量用css3实现
5. 移动端事件优化（fastclick）
6. 如何在桌面端和移动端用css开启硬件加速？（重要）

硬件加速就是将浏览器的渲染过程交给GPU处理，而不是使用自带的比较慢的渲染器，这样就可以使得animation与transition更加顺畅。我们可以在浏览器中用css开启硬件加速

.cube {

-webkit-transform: translate3d(250px,250px,250px)

rotate3d(250px,250px,250px,-120deg)

scale3d(0.5, 0.5, 0.5);

}

1. transform、translate和transition 分别是什么属性？

三者属性说明

transform 是指变换、变形，是 css3 的一个属性，和 width，height 属性一样；

translate 是 transform 的属性值，是指元素进行 2D(3D)维度上位移或范围变换;

transition 是指过渡效果，往往理解成简单的动画，需要有触发条件。

这里可以补充下 transition 和 animation 的比较，前者一般定义开始结束两个状态，需要有触发条件；

而后者引入了关键帧、速度曲线、播放次数等概念，更符合动画的定义，且无需触发条件

1. BFC理解（重要）

BFC触发条件：

1. 根元素，即html

2. float的值不为none（默认）

3. position的值为absolute或fixed

4. overflow的值不为visible（默认）

5. display的值为inline-block、table-cell、table-caption

BFC特性：

1. 内部的Box会在垂直方向上一个接一个放置。

2. Box垂直方向的距离由margin决定，属于同一个BFC的两个相邻Box的margin会发生重叠。

3. 每个元素的margin box 的左边，与包含块border box的左边相接触。

4. BFC的区域不会与float box重叠。（可用于清浮动）

5. BFC是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。

6. 计算BFC的高度时，浮动元素也会参与计算。

javaScript专题

1. js语言有什么特点？（重要）

js是一种轻量级脚本语言

弱语言类型

面向对象基于原型继承

没有静态类型检查

解释性语言

数据结构类型简单，比如只有一种数值类型

也是嵌入式语言

它的核心语法非常精简

1. 基本语法（操作符、控制结构、语句）

2. 标准库 （Array、Date、Math等）

3. 浏览器API

4. DOM

5. Web API

1. js和java的区别？（重要）

js是一门弱语言，java是一门强语言

js与Java是由不同的公司开发的不同产品

js运行在客户端浏览器，而java运行在服务端

js是解释性语言，java是编译性语言

js没有静态类型检查，java有静态类型检查

js是基于对象和事件驱动，然后使用原型继承，java是一种真正面向对象语言开发

js客户端语言不能直接对文件进行操作

1. 标识符命名规则是什么？

使用字母，\_，$,中文开头

后面可以接字母，\_,$,数字

不能使用保留关键字命名

不能使用特殊符号，空格之类

变量严格区分大小写，大小写不同则为两个变量

1. js代码引用几种方式？（重要）

三种方式，内嵌导入、外链导入和模块化导入

1. js代码插入文档中推荐放那个位置？（重要）

推荐放在</body>上面，这样可以按代码顺序下来，不会阻塞页面渲染

1. js变量有什么特点？

js变量区分大小写

变量声明会提升到代码顶部

声明变量时是先声明再赋值

如果变量是原始类型，它的值是传值传递

变量重复声明不会报错，是后面覆盖前面的

如果声明了变量而没有赋值，则它的值是undefined

var声明的变量没有局部作用域的概念

1. js有几种作用域？（必须）
2. 全局作用域
3. 函数级作用域
4. 块级作用域
5. 介绍一下你对浏览器内核的理解？（重要）

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎

渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核

JS引擎则：解析和执行javascript来实现网页的动态效果

最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎

1. 请介绍一下函数自身的作用域？

函数自身的作用域是其声明定义时所在作用域，而不是调用时的作用域

1. JS如何区分是当前是移动端还是PC端？（重要）

'ontouchend' in document;

//返回true表示移动端，false表示PC端

Reflect.has(document,'ontouchend');

//返回true表示移动端，false表示PC端

1. js代码放头部引用，并且要操作dom元素，需要注意什么？

需要添加一个window.onload事件，确保所有的dom载入完毕后在执行

1. js有那些数据类型？（必须）

数值：number

字符串：string

布尔类型：boolean

未定义：undefined

空：null

对象：object（数组array，对象object，函数function）

符号：symbol

1. js检查数据类型常用那些方法？（必须）

typeof 运算符

instanceof运算符

Object.prototype.toString 方法

1. typeof能检测那些数据类型？（必须）

数值：number

字符串：string

布尔类型：boolean

数组、对象、空：object

函数：function

未定义：undefined

符号：symbol

NaN（非数字）: number

1. 原始类型有哪几种？null 是对象嘛？（必须）

在 JS 中，存在着 6 种原始值，分别是：

boolean

null

undefined

number

string

symbol

首先原始类型存储的都是值，是没有函数可以调用的

undefined.toString();//报错

另外对于null来说，很多人会认为他是个对象类型，其实这是错误的。虽然typeof null会输出object，但是这只是 JS 存在的一个悠久 Bug。在 JS 的最初版本中使用的是 32 位系统，为了性能考虑使用低位存储变量的类型信息，000开头代表是对象，然而null表示为全零，所以将它错误的判断为object。虽然现在的内部类型判断代码已经改变了，但是对于这个 Bug 却是一直流传下来。

1. 如何最小化重绘(repaint)和回流(reflow)？（重要）
2. 需要要对元素进行复杂的操作时，可以先隐藏(display:"none")，操作完成后再显示
3. 需要创建多个DOM节点时，使用DocumentFragment创建完后一次性的加入document

缓存Layout属性值，如：var left = elem.offsetLeft; 这样，多次使用 left 只产生一次回流

1. 尽量避免用table布局（table元素一旦触发回流就会导致table里所有的其它元素回流）
2. 避免使用css表达式(expression)，因为每次调用都会重新计算值（包括加载页面）
3. 尽量使用 css 属性简写，如：用 border 代替 border-width, border-style, border-color
4. 批量修改元素样式：elem.className 和 elem.style.cssText 代替 elem.style.xxx
5. script 的位置是否会影响首屏显示时间？

在解析 HTML 生成 DOM 过程中，js 文件的下载是并行的，不需要 DOM 处理到 script 节点。因此，script的位置不影响首屏显示的开始时间。

浏览器解析 HTML 是自上而下的线性过程，script作为 HTML 的一部分同样遵循这个原则

因此，script 会延迟 DomContentLoad，只显示其上部分首屏内容，从而影响首屏显示的完成时间

1. 检测对象和数组有那些方法？（必须）

Array.isArray方法

instanceof关键字

Object.prototype.toString方法

1. null和undefine的区别？（必须）

都表示没有的意思，用法几乎一样，undefined是历史遗留问题，为了向下兼容，所以没有舍去。  
null的类型为object，undefined的类型为undefined

null转数字为0，undefined转数字为NaN 非数字

两者执行等于时转换成false

1. Base64 转码过程中的两个函数？

btoa()： ASCII 编码转换为base64

atob()： Base64 编码转换为ASCII编码

中文转换为base64编码，可以先将中文用encodeURIComponent转码为ASCII编码，然后再使用btoa转换为base64编码

1. URI 转码过程中的两个函数？

encodeURIComponent 中文转换为ASCII编码

decodeURIComponent ASCII编码转换为中文

1. 有那些值自动隐性转换为假false？（必须）

布尔假false   
数字 0   
未定义undefined   
空 null   
空字符串  
非数字 NaN

1. js存在那些数据类型转换？（必须）

强制转换

Number 、parseInt 、parseFloat 、String 和Boolean

自动（隐性）转换

123 + 'abc' // "123abc"

+ {foo: 'bar'} // NaN

true + 1 //2

1. 语句和表达式的区别？

语句

主要为了进行某种操作，一般情况下不需要返回值。

表达式

则是为了得到返回值，一定会返回一个值。

1. 那些运算结果会返回布尔值？

比较运算：> < >= <= == === != !==

逻辑运算：&& || ！

1. null 和 NaN 分别是什么类型？NaN有什么特点？（重要）

null是object类型，NaN是number类型

NaN与任何值运算都为NaN

NaN隐式转换为布尔值false

Number(null) 0

1. 0.1+0.2为什么不等于0.3，如何解决？（必须）

因为js中采用的IEEE 754的双精度标准，计算机内部存储数据的编码的时候，0.1在计算机内部根本就不是精确的0.1，而是一个有舍入误差的0.1。当代码被编译或解释后，0.1已经被四舍五入成一个与之很接近的计算机内部数字，以至于计算还没开始，一个很小的舍入错误就已经产生了。

可以先把小数转换成整数运算结束后再变为小数，如：（0.1\*10 + 0.2\*10）/ 10 == 0.3

1. Number 和 parseFloat的区别？（必须）

Number转换数字非常严格，只要包含一个字符是非数字，那么转换出来就是NaN

parseFloat它是从左向右一个一个字符查找，当发现非数字，就截止到这，Float表示浮点，所以小数位也会考虑在内。

Number转换空字符串或者null为0

parseFloat转换空字符串或者null为NaN

1. parseInt 和 Number之间的区别？

parseInt是强制转换为整数，而Number是强制转换成数值（包括浮点数）

parseInt转换规则是从带转换数据第一位开始转换，如果碰到非数字则终止转换，把已经转换的显示，而Number转换只要碰到包含非数字就输出NaN

parseInt转换的结果只有NaN和整数，而Number转换的结果有NaN、数值和0；

parseInt转换布尔值都为NaN；而Number转换true为1，false为0

parseInt转换null、空字符串和空数组为NaN；而Number转换为0

1. 数值全局方法有那些？（必须）

parseInt：强制转换成整数

parseFloat：强制转换成浮点数

Number：强制转换成数值

isNaN：判断是否为数值，返回false则为数值

1. map 和 forEach 的区别？（重要）

map 的返回值为数组，forEach的返回为undefined

map 不会改变原数组，forEach 可以改变

1. filter和some的区别？

filter 返回满足回掉函数返回true的元素数组

some 检查是否有满足回掉函数的数据，回掉函数返回true 则some返回true

1. 请简述 =、== 和 === 之间的区别？（必须）

= 表示给左侧的变量赋值

== 表示相等，会转换==左右数据类型后进行比较

=== 表示严格相等，不会转换数据类型进行比较

1. js普通（朴素）对象有那些特点？

对象是通过大括号包裹，里面以键值对的形式存在

属性之间通过英文逗号分开

键名键值之间通过英文冒号分开

键名如果为非规范命名需要用引号

键名都会自动转换成字符串

键值可以是任何类型数据

属性是无序的

键值如果是函数，称它为对象的一个方法

1. 字符串强制换行有几种办法？（重要）

有以下方法：  
1. +连接符

1. \转义符
2. es6模板字符串语法
3. 转义符是什么？请举例说明

有一些符号有特定的意义，使用转义符可以把符号当中普通没有特俗意义的符号输出。比如? 再正则里面表示0个或者1个，\?输出的就是?

1. for in 能否遍历继承的属性？

可以，所以不建议使用它遍历数组，因为这个特性使得性能下降。

如果你只想遍历对象自身的属性，可以使用

对象.hasOwnProperty(‘属性名’)

1. for for in while forEach 那些循环会跳过数组空位?（重要）

for in 、forEach会跳过数组空位

1. js有那些类数组对象？（重要）

String字符串

arguments函数参数列表对象

NodeList节点列表对象

Collection集合列表对象

FileList对象

1. 类数组对象转换为真正的数组有那些方法？（必须）

es6扩展运算符

es6方法之Array.from

Array.prototype.slice.call

1. 怎么判断一个变量没有被定义?（必须）

typeof fullname ==”undefined”

1. 添加、删除、更改、插入节点的方法（必须）

appendChild 内部后面追加

removeChild 移除子节点

replaceChild 更改替换子节点

insertBefore 在参考节点前面插入节点

1. 请写一个正则，匹配输入的字符：第一个必须是字母或下划线开头，后面就是字  
    母和数字或下划线构成，长度5-20。（必须）

/^[A-Za-z\_]\w{4,19}$/

1. 如何对一篇文章进行敏感词替换？假如有几千个敏感词。（写一下思路即可）

将这篇文章以字符串的形式赋给一个变量。定义一个正则表达式，以全局匹配的方式查找所有敏感词，并用replace方法替换掉

1. 事件委托(代理)是什么?（必须）

让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行

1. 输入URL发生什么？（必须）

1. DNS 域名解析（域名解析成ip地址，走UTP协议，因此不会有握手过程）：浏览器将 URL 解  
 析出相对应的服务器的 IP 地址（1. 本地浏览器的 DNS 缓存中查找 2. 再向系统DNS缓存发送  
 查询请求3. 再向路由器DNS缓存 4. 网络运营商DNS缓存 5. 递归搜索），并从 url 中解析出   
 端口号

2. 浏览器与目标服务器建立一条 TCP 连接（三次握手）

3. 浏览器向服务器发送一条 HTTP 请求报文

4. 服务器返回给浏览器一条 HTTP 响应报文

5. 浏览器进行渲染

6. 关闭 TCP 连接（四次挥手）

1. 浏览器渲染的步骤是什么？（必须）

1. HTML 解析出 DOM Tree

2. CSS 解析出 Style Rules

3. 两者关联生成 Render Tree

4. Layout（布局）根据 Render Tree 计算每个节点的信息

5. Painting 根据计算好的信息进行渲染整个页面

1. 页面渲染优化？（必须）

基于对渲染过程的了解，推荐一下优化：

1. HTML 文档结构层次尽量少，最好不深于 6 层 2. 脚本尽量放后边，避免组织页面加载

3. 少量首屏样式可以放在便签内

4. 样式结构层次尽量简单

5. 脚本减少 DOM 操作，减少回流，尽量缓存访问 DOM 的样式信息

6. 尽量减少 JS 修改样式，可以通过修改 class 名的方式解决

7. 减少 DOM 查找，缓存 DOM 查找结果

8. 动画在屏幕外或页面滚动时，尽量停止

1. session 与 cookie 的区别？（重要）

session 保存在服务器，客户端不知道其中的信息；

cookie 保存在客户端，服务器能够知道其中的信息。

session 中保存的是对象，cookie 中保存的是字符串。

session 不能区分路径，同一个用户在访问一个网站期间，所有的 session 在任何一个地方都可以访问到。

而 cookie 中如果设置了路径参数，那么同一个网站中不同路径下的 cookie 互相是访问不到的。

1. 强制缓存和协商缓存？（重要）

强制缓存是我们在第一次请求资源时在 http 响应头设置一个过期时间，在时效内都将直接从浏览

器进行获取，常见的 http 响应头字段如 Cache-Control 和 Expires

协商缓存是我们通过 http 响应头字段 etag 或者 Last-Modified 等判断服务器上资源是否修改，如果修改则从服务器重新获取，如果未修改则 304 指向浏览器缓存中进行获取

1. 描述下JSON对象的两个很重要的方法？（必须）

JSON.parse() //JSON字符串转换为JSON对象

JSON.stringify() //JSON对象转化为字符串

1. 检测浏览器的两种方式？（必须）

功能检测 “touchstart” in document

特征检测 navigator.userAgent

1. “===”和“ ==”的区别？（必须）

==，当且仅当两个运算数相等时，它返回 true，即不检查数据类型

===，只有在无需类型转换运算数就相等的情况下，才返回 true，需要检查数据类型

1. 什么是原型和原型链？（必须）

每个由构造函数new出来的实例化对象都自带一个\_proto\_属性，该属性指向创建它的构造函数的prototype对象。而prototype对象因为是实例，也有自己的\_proto\_属性，指向它的原型对象。当我们访问一个对象的属性时，如果这个对象内部不存在这个属性，那么他就会去prototype里找这个属性，这个prototype又会有自己的prototype，

于是就这样一直找下去，也就是我们平时所说的原型链的概念。

关系：instance.constructor.prototype = instance.\_\_proto\_\_

1. cookies，sessionStorage 和 localStorage 的区别？（必须）

cookie是网站为了标示用户身份而储存在用户本地终端（Client Side）上的数据（通常经过加密）。

cookie数据始终在同源的http请求中携带（即使不需要），记会在浏览器和服务器间来回传递。

sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。

存储大小

cookie数据大小不能超过4k。

sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。

生命周期

localStorage 存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；

sessionStorage 数据在当前浏览器窗口关闭后自动删除。

cookie 设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭

1. 什么叫优雅降级和渐进增强？（重要）

优雅降级

Web站点在所有新式浏览器中都能正常工作，如果用户使用的是老式浏览器，则代码会针对旧版本的IE进行降级处理了,使之在旧式浏览器上以某种形式降级体验却不至于完全不能用。

如：border-shadow

渐进增强

从被所有浏览器支持的基本功能开始，逐步地添加那些只有新版本浏览器才支持的功能,向页面增加不影响基础浏览器的额外样式和功能的。当浏览器支持时，它们会自动地呈现出来并发挥作用。

如：默认使用flash上传，但如果浏览器支持 HTML5 的文件上传功能，则使用HTML5实现更好的体验；

1. js和ecma的关系？（重要）

ecma是js的制定标准

js是ecma的一种实现，除此还有jscript、actionscript等

1. eval是做什么的？

它的功能是把对应的字符串解析成JS代码并运行；

应该避免使用eval，不安全，非常耗性能（2次，一次解析成js语句，一次执行）。

由JSON字符串转换为JSON对象的时候可以用eval，var obj =eval('('+ str +')');

1. 说几条写JavaScript的基本规范？（必须）

（1）不要在同一行声明多个变量。

（2）请使用 ===/!==来比较true/false或者数值

（3）使用对象字面量替代new Array这种形式

（4）不要使用全局函数。

（5）switch语句必须带有default分支

（6）函数不应该有时候有返回值，有时候没有返回值。

（7）for循环必须使用大括号

（8）if语句必须使用大括号

（9）for-in循环中的变量 应该使用var关键字明确限定作用域，从而避免作用域污染。

1. ["1", "2", "3"].map(parseInt) 答案是多少？（必须）

[1, NaN, NaN] 因为 parseInt 需要两个参数 (val, radix)，

其中 radix 表示解析时用的基数。

map 传了 3 个 (element, index, array)，对应的 radix 不合法导致解析失败。

1. 阻止默认动作和阻止冒泡？（必须）

阻止默认动作

w3c的方法是e.preventDefault()，IE则是使用e.returnValue = false;

阻止冒泡

w3c的方法是e.stopPropagation()，IE则是使用e.cancelBubble = true

函数中return false

原生js的return false只会阻止默认行为，而是用jQuery的话则既阻止默认行为又防止对象冒泡。

1. 介绍一下rAF(requestAnimationFrame)？(了解)

专门用来做动画，不卡顿，用法和setTimeout一样。  
定时器一直是 js 动画的核心技术，但它们不够精准，因为定时器时间参数是指将执行代码放入 UI 线程队列中等待的时间，如果前面有其他任务队列执行时间过长，则会导致动画延迟，效果不精确等问题。所以处理动画循环的关键是知道延迟多长时间合适：时间要足够短，才能让动画看起来比较柔滑平顺，避免多余性能损耗；时间要足够长，才能让浏览器准备好变化渲染。这个时候 rAF 就出现了，采用系统时间间隔(大多浏览器刷新频率是 60Hz，相当于 1000ms/60≈16.6ms)，保持最佳绘制效率，不会因为间隔时间过短，造成过度绘制，增加开销；也不会因为间隔时间太长，使用动画卡顿不流畅，让各种网页动画效果能够有一个统一的刷新机制。并且 rAF 会把每一帧中的所有 DOM 操作集中起来，在一次重绘或回流中就完成。

1. javascript 的垃圾回收机制讲一下 (了解)

定义：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。

像 C 这样的编程语言，具有低级内存管理原语，如 malloc()和 free()。开发人员使用这些原语显式地对操作系统的内存进行分配和释放。

而 JavaScript 在创建对象(对象、字符串等)时会为它们分配内存，不再使用对时会“自动”释放内存，这个过程称为垃圾收集。

内存生命周期中的每一个阶段：

分配内存 — 内存是由操作系统分配的，它允许您的程序使用它。在低级语言(例如 C 语言)中，这是一个开发人员需要自己处理的显式执行的操作。然而，在高级语言中，系统会自动为你分配内在。

使用内存 — 这是程序实际使用之前分配的内存，在代码中使用分配的变量时，就会发生读和写操作。

释放内存 — 释放所有不再使用的内存,使之成为自由内存,并可以被重利用。与分配内存操作一样,这一操作在低级语言中也是需要显式地执行。

1. js常见的常见的内存泄漏（重要）

四种常见的内存泄漏：

1. 全局变量 不用 var 声明的变量，相当于挂载到 window 对象上。如：b=1; 解决：  
使用严格模式

2. 被遗忘的定时器和回调函数

3. 闭包

4. 没有清理的 DOM 元素引用

1. 对前端性能优化有什么了解？一般都通过那几个方面去优化的？（必须）

1. 减少请求数量

2. 减小资源大小

3. 优化网络连接

4. 优化资源加载

5. 减少重绘回流

6. 性能更好的API

7. webpack优化

1. 函数的声明方式有几种？区别是什么？（重要）

函数的声明方式有三种

语句式声明：函数可提升，必须有函数名

表达式声明：函数不可提升，可以有函数名，也可以没有

构造函数式声明：函数不可提升，最后一个参数是函数体，会将字符串解析为表达式

1. return只能在函数内部使用？return有什么作用？

是的，只能在函数内部使用。return会终止函数的运行，不再执行后面的语句，然后返回函数后面的那个表达式。

1. break和continue的区别？（重要）

都是用在for或者while循环语句，break还可以用在switch语句中

break表示跳出循环，continue表示跳过本次循环，然后继续。

1. 函数有那些作用域？（必须）

全局作用域

局部作用域

块级作用域

1. 函数参数有什么特点？

1. 参数没有个数限制

1. 函数参数分实参和形参，形参可以自定义但要符合标识符命名规则，形参和实参要一一对应
2. 函数可以没有参数

4 .函数参数省略规则是从后往前的，不能省略中间的参数，不可跳跃式省略

1. 函数的参数值可以是任何类型
2. 函数参数若是原始类型执行传值传递、函数参数若是引用类型执行传址传递
3. 函数形参若存在同名的情况，取最后一个参数，并且形参可以设置默认值
4. 闭包的概念？优缺点？（必须）

闭包的概念：闭包就是能读取其他函数内部变量的函数。

优点

1. 避免全局变量的污染

2. 希望一个变量长期存储在内存中（缓存变量）

缺点

1. 内存泄露（消耗）

2. 常驻内存，增加内存使用量

1. 从输入url到显示页面，都经历了什么？（必须）
2. 首先，在浏览器地址栏中输入url
3. 浏览器先查看浏览器缓存-系统缓存-路由器缓存，如果缓存中有，会直接在屏幕中显示页面内容。  
    若没有，则跳到第三步操作。
4. 在发送http请求前，需要域名解析(DNS解析)(DNS（域名系统，Domain Name System）是互  
    联网的一项核心服务，它作为可以将域名和IP地址相互映射的一个分布式数据库，能够使人更方  
    便的访问互联网，而不用去记住IP地址。)，解析获取相应的IP地址。
5. 浏览器向服务器发起tcp连接，与浏览器建立tcp三次握手。（TCP即传输控制协议。TCP连接  
    是互联网连接协议集的一种。）
6. 握手成功后，浏览器向服务器发送http请求，请求数据包。
7. 服务器处理收到的请求，将数据返回至浏览器。
8. 浏览器收到HTTP响应。
9. 读取页面内容，浏览器渲染，解析html源码。
10. 生成Dom树、解析css样式、js交互。
11. 客户端和服务器交互。
12. ajax查询。
13. 函数声明方式有几种？

语句式声明

表达式声明

构造函数式声明

1. 函数语句式和表达式声明有什么区别？

1. 定义的方式不同，一个是语句式，一个表达式。

2. 函数声明式必须要有函数名，而函数表达式可以不需要函数名，它是匿名函数。

3. 使用函数语句式声明，函数会自动提升到代码的顶端，而使用函数表达式声明，则不会自动提升。

1. 函数 return 关键字作用？

return 是函数内部的关键字，不能在函数外部使用

它的用途是在函数内部中返回值并跳出函数

1. IIFE是什么？有什么作用？有几种写法？（必须）

IIFE是立即调用函数表达式。

作用：

1. 函数立即执行

2. 由于使用匿名函数，不必为函数命名，避免为命名烦恼；

3. 由于代码写在函数内部，所以具有函数级作用域，避免污染全局变量。

4. 可以拿它这种结构来封装插件

写法有以下几种：

+function(){}  
~function(){}  
!function(){}  
-function(){}  
(function(){})  
(function(){})()

1. use strict和普通模式有什么区别？（必须）

use strict 表示严格模式。

严格模式下，为了代码安全，禁止在函数内部遍历argmenuts.callee

严格模式下，为了代码安全，禁止this指向全局对象window

普通模式下不写var关键字声明赋值变量不会报错，严格模式下会报错

严格模式下去掉了with关键字；

严格模式下eval本身就是一个作用域，而普通模式下是全局作用域或者函数级作用域

1. 三个数字之间判断最大值的方法？

//方法1  
console.log(Math.max(1,2,3))  
  
//方法2  
var a = 1;  
var b = 2;  
var c = 3;  
var max = a>b?(a>c?a:c):(b>c?b:c);  
console.log(max);

1. ++x和x++的区别是什么？（必须）

++x是先加后输出

x++ 是先输出后加

1. js数据类型隐式转换有那些？（至少5种不同类型）（必须）

类型一：true+true//输出2

类型二：1+’1’//输出11

类型三：null == undefined //输出true  
类型四：‘2’>1 //输出true

类型五：!’ab’ //输出false

1. js强制数据类型转换有那些？（必须）

parseInt 强制转换为整型

Number强制转换为数值型

parseFloat强制转换为浮点型

Boolean 强制转换为布尔

String 强制转换为字符串

1. typeof undefined == typeof NULL的值输出什么？（重要）

true，js变量区分大小写 NULL不存在 左边和又变都是输出为undefined，所以返回true

1. js原生错误有那些？（重要）

SyntaxError 语法错误

ReferenceError 引用错误

RangeError 范围错误、

TypeError 类型错误

URIError URI编码错误

EvalError 编码错误

1. DOM 事件有哪些阶段？谈谈对事件代理的理解？（必须）

分为三大阶段：捕获阶段--目标阶段--冒泡阶段

事件代理简单说就是：事件不直接绑定到某元素上，而是绑定到该父元素上，进行触发事件操作

时(例如'click')，再通过条件判断，执行事件触发后的语句(例如'alert(e.target.innerHTML)')

好处：(1)使代码更简洁；(2)节省内存开销

1. 什么是事件流？（必须）

DOM(文档对象模型)结构是一个树型结构，当一个HTML元素产生一个事件时，该事件会在元素结点与根结点之间的路径传播，路径所经过的结点都会收到该事件，这个传播过程可称为DOM事件流。DOM同时支持两种事件模型：捕获型事件和冒泡型事件

1. JavaScript原型继承是如何运作的？（必须）

每个对象都会在其内部初始化一个属性，就是prototype(原型)，当我们访问一个对象的属性时，如果这个对象内部不存在这个属性，那么他就会去prototype里找这个属性，这个prototype又会有自己的prototype，于是就这样一直找下去(知道找到null为止)，也就是我们平时所说的原型链的概  
念。

1. CORS是什么？对于跨域请求，如何将附带凭证（HTTP Cookie 及 HTTP认证信息）的请求发送至服务器端?（必须）

CORS是一个W3C标准，全称是"跨域资源共享"（Cross-origin resource sharing）。

它允许浏览器向跨源服务器，发出XMLHttpRequest请求，从而克服了AJAX只能同源使用的限制。只需要服务器端的Response header 设置下Access-Control-Allow-Origin即可。

只需把相应的cookie信息和认证信息作为post请求发送给服务器端即可。

1. 什么是http协议？（重要）

HTTP协议（HyperText Transfer Protocol，超文本传输协议）是用于从WWW服务器传输超文本到本地浏览器的传送协议。

HTTP是一个应用层协议，由请求和响应构成，是一个标准的客户端服务器模型，HTTP是一个无状态的协议。

默认HTTP的端口号为80，HTTPS的端口号为443。

HTTP协议永远都是客户端发起请求，服务器回送响应。

HTTP的常见请求方法有两种GET和POST。

HTTP工作流程：一次HTTP操作称为一个事务，其工作过程可分为四步

首先客户机与服务器需要建立连接。只要单击某个超级链接，HTTP的工作开始。

建立连接后，客户机发送一个请求给服务器，请求方式的格式为：统一资源标识符（URL）、协议版本号，后边是MIME信息包括请求修饰符、客户机信息和可能的内容。

服务器接到请求后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。

客户端接收服务器所返回的信息通过浏览器显示在用户的显示屏上，然后客户机与服务器断开连接。

如果在以上过程中的某一步出现错误，那么产生错误的信息将返回到客户端，有显示屏输出。对于用户来说，这些过程是由HTTP自己完成的，用户只要用鼠标点击，等待信息显示就可以了。

1. HTTPS 与 HTTP 相比？

HTTPS需要证书

HTTPS协议需要到CA申请证书，一般免费证书很少，需要交费。

HTTP协议运行在TCP之上，所有传输的内容都是明文，HTTPS运行在SSL/TLS之上，SSL/TLS运行在TCP之上，所有传输的内容都经过加密的。

端口不同

HTTP和HTTPS使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。

HTTPS可以有效的防止运营商劫持，解决了防劫持的一个大问题。

HTTPS在传输数据之前需要客户端（浏览器）与服务端（网站）之间进行一次握手，在握手过程中将确立双方加密传输数据的密码信息。TLS/SSL协议不仅仅是一套加密传输的协议，TLS/SSL中使用了非对称加密，对称加密以及HASH算法。

1. 什么是构造函数？（必须）

js中构造函数的定义和普通函数的定义是一样的，如何确定是否构造函数的关键在于是否使用了new关键字。使用了new关键字的普通函数，我们称这个函数为构造函数。

1. new 操作符具体干了什么？（必须）

new 共经历了五个过程。

1. 创建一个空对象，作为将要返回的对象实例
2. 将这个空对象的原型，指向构造函数的prototype属性
3. 将这个空对象赋值给函数内部的this关键字
4. 开始执行构造函数内部的代码。
5. 如果返回来的是对象则输出这个对象，否则输出上面的空对象
6. 用过哪些设计模式？（必须）

工厂模式：简单的工厂模式可以理解为解决多个相似的问题  
单例模式：只能被实例化(构造函数给实例添加属性与方法)一次

沙箱模式： 将一些函数放到自执行函数里面,但要用闭包暴露接口,用变量接收暴露的接口,再调用里面的值,否则无法使用里面的值

发布者订阅模式：就例如如我们关注了某一个公众号,然后他对应的有新的消息就会给你推送

1. use strict的主要目的？（重要）

明确禁止一些不合理、不严谨的语法，减少 JavaScript 语言的一些怪异行为。

增加更多报错的场合，消除代码运行的一些不安全之处，保证代码运行的安全。

提高编译器效率，增加运行速度。

为未来新版本的 JavaScript 语法做好铺垫。

1. DOM元素Attribute与Property的区别是什么？

property是DOM中的属性，是JavaScript里的对象；而attribute是HTML标签上的特性，它的值只能够是`字符串；

DOM对象初始化时会在创建默认的基本property；只有在HTML标签中定义的attribute才会被保存在property的attributes属性中；

attribute会初始化property中的同名属性，但自定义的attribute不会出现在property中；

propety是对象，而attribute的值都是字符串；

最关键的两句话：

attribute（特性），是我们赋予某个事物的特质或对象。

property（属性），是早已存在的不需要外界赋予的特质。

1. ajax的原理是什么？（必须）

Ajax基于XMLHttpRequest对象与Web服务器端进行异步数据通信。

首先基于这个对象的open方法创建一个浏览器跟服务器端连接，通过send方法从浏览器向服务器端发送请求。我们可以通过这个对象的onreadystatechange事件来监听请求的状态，当请求成功之后的话，我可以获取到这个对象responseText等方法获取到请求过来的数据，然后通过js对这些数据进行解析。

1. 简述DOM，HTMLDOM的区别和联系？

DOM分为三部分：

（1）核心DOM：遍历DOM树、添加新节点、删除节点、修改节点 （2）HTML DOM：以一种简便的方法访问DOM树 （3）XML DOM：准用于操作XML文档

核心DOM与HTML DOM的区别：

核心DOM

对象：Document,Node, ElementNode, TextNode,AttributeNode，CommentNode, NodeList 核心DOM提供了统一的操作接口，如：createElement、appendChild、setAttribute等 核心DOM创建新元素：var newNode = document.createElement("img")

给元素添加属性：e.setAttribure()、e.setAttribureNode() 适用场合：核心DOM适合操作节点，如创建，删除，查找等

HTML DOM

对象：image, Table, Form, Input,Select等等HTML标签对象化 HTML DOM提供了封装好的各种对象，如：Select、Option等等 适用场合：HTML DOM适合操作属性，如读取或修改属性的值

1. Set是什么，有什么作用？

Set是ES6引入的一种类似Array的新的数据结构，Set实例的成员类似于数组item成员，区别是Set实例的成员都是唯一，不重复的。这个特性可以轻松地实现数组去重

1. Map是什么，有什么作用？（重要）

Map是ES6引入的一种类似Object的新的数据结构，Map可以理解为是Object的超集，打破了以传统键值对形式定义对象，对象的key不再局限于字符串，也可以是Object。可以更加全面的描述对象的属性

1. let ，const ，var 有什么区别？（必须）

什么时候提出的？

var是ES5提出的，let和const是ES6提出的。

是否存在变量提升？

var声明的变量存在变量提升

let和const不存在变量提升

是否存在暂时性死区？

let和const存在暂时性死区。即只要块级作用域内存在let命令，它所声明的变量就“绑定”（binding）这个区域，不再受外部的影响。

在代码块内，使用let命令声明变量之前，该变量都是不可用的。这在语法上，称为“暂时性死区”

是否允许重复声明变量？

var允许重复声明变量。let和const在同一作用域不允许重复声明变量。

是否存在块级作用域？

var不存在块级作用域。let和const存在块级作用域。

是否能修改声明的变量？

var和let可以, const不可以。

1. 异步加载Js的方式有哪些？（必须）

script 标签的 async 属性，兼容性IE9+

script 的 defer 属性，兼容所有的浏览器  
动态创建script标签，兼容所有的浏览器  
ajax加载脚本内容，兼容所有的浏览器

iframe方式，兼容所有的浏览器

1. js有那些节点类型？

Document 文档节点（nodeType=9）

DocumentType 文档类型节点（nodeType=10）

Element 元素节点（nodeType=1）

Attribute 属性节点（nodeType=2）

Text 文本节点（nodeType=3）

Comment 注释节点（nodeType=8）

DocumentFragment 文档片段（nodeType=11）

1. 节点有那些关系？

parentNode 父节点关系

childNodes 子节点关系

sibling 同级节点关系

1. nodelist 对象和collection的区别？
2. 都是一种数据结构，类数组对象。
3. collection成员必须是元素，而nodelist是节点类型
4. nodelist有forEach方法，而collection没有
5. onload 和 DOMContentLoaded 区别？

DOMContentLoaded 比onload事件发生要早

DOMContentLoaded 无需等待样式表、图像和子框架（iframe）的完成加载

onload页面的html、css、js、图片等资源都已经加载完之后才会触发 load 事件

1. click、mousedown、mouseup事件触发的顺序是？（重要）

mousedown > mouseup > click

1. mouseenter和mouseover的区别？（重要）

当鼠标移动到元素上时就会触发mouseenter事件

类似mouseover，它们两者之间的差别是：

mouseover

鼠标经过自身盒子会触发，经过子盒子还会触发（因为存在冒泡）

mouseenter

只会经过自身盒子触发（不会冒泡）

跟mouseenter对应的，鼠标离开事件mouseleave同样不会冒泡

1. currentTarget、relatedTarget和target属性的区别？（重要）

target指的你触发目标对象，这个对象可能是后代元素，它不一定是你绑定事件的对象

relatedTarget 通过这个属性可以得到上一次触发的对象

currentTarget 指的是你绑定事件的对象

1. clientX、screenX和pageY的区别？（重要）

clientX相对于窗口可视宽度的水平坐标

screenX相对屏幕的水平坐标

pageY相对于窗口（含溢出部分）的水平坐标

1. 怎么添加、移除、复制、创建、和查找节点？（必须）

创建新节点

createElement() //创建一个具体的元素

createTextNode() //创建一个文本节点

createDocumentFragment() //创建一个DOM片段

添加、移除、替换、之前插入、之后插入、复制

appendChild() //内部后面追加节点

removeChild() //删除子节点

replaceChild() //替换子节点

insertBefore() //在已有的子节点前插入一个新的子节点

insertAfter() //方法在被选元素后插入 `HTML` 元素。

cloneNode() //复制节点

查找

document.getElementsByTagName('div'); // 通过标签名称

document.getElementsByName('hobby'); // 通过元素的Name属性的值

document.getElementById('gender'); // 通过元素Id，唯一性

document.getElementsByClassName('red'); // 通过类查找

document.querySelector('.red'); //通过css类型查找

1. 在css/js代码上线之后开发人员经常会优化性能，从用户刷新网页开始，一次  
    js请求一般情况下有哪些地方会有缓存处理 (了解)

dns缓存

cdn缓存

浏览器缓存

服务器缓存

1. 如何处理HTML5新标签的浏览器兼容问题？ (了解)

IE8/IE7/IE6支持通过 document.createElement 方法产生的标签，可以利用这一特性让这些浏览器支持HTML5 新标签，浏览器支持新标签后，还需要添加标签默认的样式（当然最好的方式是直接使用成熟的框架、使用最多的是html5shim框架）：

<!--[if lt IE 9]>

<script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>

<![endif]-->

1. 简述同步和异步的区别?（重要）

同步是阻塞模式，异步是非阻塞模式。

同步就是指一个进程在执行某个请求的时候，若该请求需要一段时间才能返 回信息，那么这个进程将会一直等待下去，直到收到返回信息才继续执行下去。

异步是指进程不需要一直等下去， 而是继续执行下面的操作，不管其他进程的状态。当有消息返回时系统会通知进程进行处理，这样可以提高执行的效率。

1. 如何改变函数内部的this指针的指向？（必须）

call方法

apply方法

bind方法

es6 箭头函数

1. 跨域有那些办法，请说明原理？（必须）

jsonp

利用 <script> 标签没有跨域限制的漏洞，网页可以得到从其他来源动态产生的 JSON 数据。JSONP请求一定需要对方的服务器做支持才可以。

CORS

CORS 需要浏览器和后端同时支持。IE 8 和 9 需要通过 XDomainRequest 来实现。

postMessage

postMessage()方法允许来自不同源的脚本采用异步方式进行有限的通信，可以实现跨文本档、多窗口、跨域消息传递。

websocket

Websocket是HTML5的一个持久化的协议，它实现了浏览器与服务器的全双工通信，同时也是跨域的一种解决方案。WebSocket和HTTP都是应用层协议，都基于 TCP 协议。但是 WebSocket 是一种双向通信协议，在建立连接之后，WebSocket 的 server 与 client 都能主动向对方发送或接收数据。

Node中间件代理

实现原理：同源策略是浏览器需要遵循的标准，而如果是服务器向服务器请求就无需遵循同源策略。

window.name + iframe

window.name属性的独特之处：name值在不同的页面（甚至不同域名）加载后依旧存在，并且可以支持非常长的 name 值（2MB）。

总结

CORS支持所有类型的HTTP请求，是跨域HTTP请求的根本解决方案

JSONP只支持GET请求，JSONP的优势在于支持老式浏览器，以及可以向不支持CORS的网站请求数据。

不管是Node中间件代理还是nginx反向代理，主要是通过同源策略对服务器不加限制。

日常工作中，用得比较多的跨域方案是cors和nginx反向代理

1. 原型和原型链，原型的作用？（必须）

原型

所有引用类型都有一个\_\_proto\_\_(隐式原型)属性，属性值是一个普通的对象。

所有函数都有一个prototype(原型)属性，属性值是一个普通的对象。

prototype对象有一个constructor属性，默认指向prototype对象所在的构造函数。由于constructor属性定义在prototype对象上面，意味着`可以被所有实例对象继承。

所有引用类型的\_proto\_属性指向它构造函数的prototype。

原型链

当访问一个对象的某个属性时，会先在这个对象本身属性上查找，如果没有找到，则会去它的\_proto\_隐式原型上查找，即它的构造函数的prototype，如果还没有找到就会再在构造函数的prototype的\_proto\_中查找，这样一层一层向上查找就会形成一个链式结构，我们称为原型链。

原型的作用

数据共享，节约内存内存空间。

实现继承。

1. 至少说出三种异步操作的使用场景 （必须）

回调函数

事件监听

Promise对象

Nodejs的异步方法

1. 常见web安全及防护原理（必须）

sql注入原理

通过把sql命令插入到web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意sql命令

xss攻击

跨站脚本是一种网站应用程序的安全漏洞攻击，是代码注入的一种。它允许恶意用户将代码注入到网页上，其他用户在观看网页时就会受到影响。这类攻击通常包含了HTML以及用户端脚本语言。

比如：攻击者在论坛中放一个看似安全的链接，骗取用户点击后，窃取Cookie中的用户私密信息；或者攻击者在论坛中加一个恶意表单，当用户提交表单的时候，却把信息传送到攻击者的服务器中，而不是用户原本以为的信任站点。

sql注入和xss攻击防范

1. 首先代码里对用户输入的地方和变量都需要仔细检查长度和对< , > , ; , ’等字符做过滤；其次任何内容写到页面之前都必须加以encode，避免不小心把html tag弄出来。这一个层面做好，至少可以堵住超过一半的XSS攻击。
2. 永远不要使用动态拼装SQL，可以使用参数化的SQL或者直接使用存储过程进行数据查询存取。
3. 永远不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的数据库连接。
4. 不要把机密信息明文存放，请加密或者hash掉密码和敏感的信息

CSRF

跨站请求伪造，是一种挟制用户在当前已登录的Web应用程序上执行非本意的操作的攻击方法。

就是说冒充用户发起请求（在用户不知情的情况下）,完成一些违背用户意愿的请求（如恶意发帖，删帖，改密码，发邮件等）。只要是伪造用户发起的请求，都可成为CSRF攻击。

CSRF防范

服务端的CSRF方式方法很多样，但总的思想都是一致的，就是在客户端页面增加伪随机数，通过验证码的方法。

XSS与CSRF的区别

XSS是获取信息，不需要提前知道其他用户页面的代码和数据包。CSRF是代替用户完成指定的动作，需要知道其他用户页面的代码和数据包。

要完成一次CSRF攻击，受害者必须依次完成两个步骤：

a. 登录受信任网站A，并在本地生成Cookie。

b. 在不登出A的情况下，访问危险网站B。

1. 为什么要同源限制？（重要）

同源策略指的是协议，域名，端口相同，同源策略是一种安全协议。

同源策略的本质是一种约定，可以说web的行为就是构建在这种约定之上的。就好比我们人类的行为必须受到法律的约束一样，同源策略的目的就是限制不同源的document或者脚本之间的相互访问，以免造成干扰和混乱、隐私泄漏。

1. 怎么确定是同源？（重要）

协议相同

域名相同

端口相同

1. 怎么算跨域？那些不会跨域？（重要）

只要协议、域名或者端口有一个不同就算跨域，

script标签src引入网址不受同源策略限制

后端也不受同源策略限制

1. 非同源（跨域）有那些限制？（重要）

无法读取非同源网页的 Cookie、LocalStorage 和 IndexedDB

无法接触非同源网页的 DOM

无法向非同源地址发送 AJAX 请求

1. ES5 和 ES6 分别几种方式声明变量?（重要）

ES5 有俩种： var 和 function

ES6 有六种：增加四种， let 、 const 、 class 和 import

注意： let 、 const 、 class 声明的全局变量再也不会和全局对象的属性挂钩

1. 有几种发送http请求的方式？
2. 地址栏输入网址，发送了http请求（GET）
3. 通过form发送请求，请求类型不限
4. 通过超级链接跳转，发送了http请求（GET）

4. js编程式发送http请求

a. 通过脚本重定向（location.href）

b. 通过ajax发送请求（从后台的方式通讯）

5. 通过后端发送http请求（常说采集、爬虫程序）

1. offsetWidth,clientWidth与scrollWidth的区别？（重要）

offsetWidth返回值包含content + padding + border ，如果有滚动条，也不包含滚动条。

clientWidth返回值只包含content + padding，如果有滚动条，也不包含滚动条。

scrollWidth返回值包含content + padding + 溢出内容的尺寸

1. http常见状态码有那些？（必须）

200 服务器已成功处理了请求

301 （永久移动） 请求的网页已永久移动到新位置

302 （临时移动） 服务器目前从不同位置的网页响应请求。

304 （未修改） 自从上次请求后，请求的网页未修改过。

307 （临时重定向） 服务器目前从不同位置的网页响应请求

401 （未授权） 请求要求身份验证

403 （禁止） 服务器拒绝请求

404 （未找到） 服务器找不到请求的网页

406 （不接受） 无法使用请求的内容特性响应请求的网页

500 （服务器内部错误） 服务器遇到错误，无法完成请求。

502 （错误网关） 服务器作为网关或代理，从上游服务器收到无效响应。

503 （服务不可用） 服务器目前无法使用（由于超载或停机维护）

1. javascript有哪些方法定义对象?

创建Object的方式有4种。

通过对象字面量表示法（又称为直接量、原始方式）。

var obj = {name:'张三'};

通过new和构造函数Object()

var obj = new Object();

对象的构造函数，然后实例化对象

function obj(o){

this.name = '张三';

}

var obj = new obj();

通过Object.create()

var o1 = Object.create({x:1, y:2});

1. 文件请求文档类型（Content-Type）有那些？（重要）

application/x-www-form-urlencoded

application/json

multipart/form-data

text/plain

1. xhr对象中withCredentials属性有什么用？

属性是一个布尔值，表示跨域请求时，设置为true，用户信息例如Cookie 会被发送给后端

为了让这个属性生效，服务器必须显式返回 Access-Control-Allow-Credentials: true

1. 说说你对promise的了解? （重要）

Promise是什么？

Promise是最早由社区提出和实现的一种解决异步编程的方案，比其他传统的解决方案（回调函数和事件）更合理和更强大。

ES6 将其写进了语言标准，统一了用法，原生提供了Promise对象。 ES6 规定，Promise对象是一个构造函数，用来生成Promise实例。

Promise是为解决什么问题而产生的？

promise是为解决异步处理回调金字塔问题（回调噩梦）而产生的

Promise的两个特点

1、Promise对象的状态不受外界影响

1）pending 初始状态

2）fulfilled 成功状态

3）rejected 失败状态

Promise 有以上三种状态，只有异步操作的结果可以决定当前是哪一种状态，其他任何操作都无法改变这个状态

2、Promise的状态一旦改变，就不会再变，任何时候都可以得到这个结果，状态不可以逆，只能由 pending变成fulfilled或者由pending变成rejected

Promise的三个缺点

1）无法取消Promise，一旦新建它就会立即执行，无法中途取消   
2）如果不设置回调函数，Promise内部抛出的错误，不会反映到外部   
3）当处于pending状态时，无法得知目前进展到哪一个阶段，是刚刚开始还是即将完成

Promise的用法

1. Promise构造函数接受一个函数作为参数，该函数的两个参数分别是resolve和reject。它们是两个函数，由 JavaScript 引擎提供，不用自己部署。

const promise = new Promise(function(resolve, reject) {

// ... some code

if (/\* 异步操作成功 \*/){

resolve(value);

} else {

reject(error);

}

});

2、resolve函数的作用是，将Promise对象的状态从“未完成”变为“成功”（即从 pending 变为 resolved），在异步操作成功时调用，并将异步操作的结果，作为参数传递出去；reject函数的作用是，将Promise对象的状态从“未完成”变为“失败”（即从 pending 变为 rejected），在异步操作失败时调用，并将异步操作报出的错误，作为参数传递出去。

3、 Promise实例生成以后，可以用then方法分别指定resolved状态和rejected状态的回调函数。

promise.then(function(value) {

// success

}, function(error) {

// failure

});

then方法可以接受两个回调函数作为参数。第一个回调函数是Promise对象的状态变为resolved时调用，第二个回调函数是Promise对象的状态变为rejected时调用。其中，第二个函数是可选的，不一定要提供。这两个函数都接受Promise对象传出的值作为参数。

Promise 链式写法

我们先来看一个例子，根据例子得出结论，然后再写源码的实现部分来验证结论

let promise = new Promise(function (resolve, reject) {

resolve(100);// reject(100)

});

promise.then(function (data) {

return data+100;

},function (err) {

return 'ssss';

}).then(function (data) {

console.log(data);// 200 // undefined // sss

})

从上面的例子可以看出:

当第一个promise的成功的回调里返回 200时，第二个promise的成功回调的参数就是200 当将resolve(100)改成reject(100)的时候，因为失败回调中什么也没有返回所以第二个promise的成功回调中的参数是undefined 当失败的回调中返回sss时，第二个promise的成功回调中的参数是sss

由此我们可以看出，第一个promise不管成功回调还是失败回调，他的返回值作为第二个promise中的成功时回调函数的参数值

链式写法能一直then下去的原因：链式调用靠的是返回新的promise，来保证可以一直走成功或失败

Promise.catch

Promise.prototype.catch方法是.then(null, rejection)的别名，用于指定发生错误时的回调函数。

//catch原理就是只传失败的回调

Promise.prototype.catch = function(onRejected){

this.then(null,onRejected);

}

Promise.all 方法

参数：接受一个数组，数组内都是Promise实例 返回值：返回一个Promise实例，这个Promise实例的状态转移取决于参数的Promise实例的状态变化。当参数中所有的实例都处于resolve状态时，返回的Promise实例会变为resolve状态。如果参数中任意一个实例处于reject状态，返回的Promise实例变为reject状态

Promise.resolve

返回一个Promise实例，这个实例处于resolve状态。 根据传入的参数不同有不同的功能：

值(对象、数组、字符串等)：作为resolve传递出去的值 Promise实例：原封不动返回

Promise.reject

返回一个Promise实例，这个实例处于reject状态。

参数一般就是抛出的错误信息。

Promise.race

参数：接受一个数组，数组内都是Promise实例 返回值：返回一个Promise实例，这个Promise实例的状态转移取决于参数的Promise实例的状态变化。当参数中任何一个实例处于resolve状态时，返回的Promise实例会变为resolve状态。如果参数中任意一个实例处于reject状态，返回的Promise实例变为reject状态。

1. Web开发中会话跟踪的方法有哪些？（重要）

1、cookie

2、session

3、隐藏input

4、url重写

5、ip地址

1. 介绍js有哪些内置对象？（重要）

Object 是 JavaScript 中所有对象的父对象。

数据封装类对象

Object

Array

Boolean

Number

String

Regex

其他对象

Function

Arguments

Math

Date

RegExp

Error

JSON

1. 说说严格模式的限制？

说说严格模式的限制

变量必须声明后再使用

函数的参数不能有同名属性，否则报错

不能使用with语句

不能对只读属性赋值，否则报错

不能使用前缀0表示八进制数，否则报错

不能删除不可删除的属性，否则报错

不能删除变量delete prop，会报错，只能删除属性delete global[prop]

eval不会在它的外层作用域引入变量

eval和arguments不能被重新赋值

arguments不会自动反映函数参数的变化

不能使用arguments.callee

不能使用arguments.caller

禁止this指向全局对象

不能使用fn.caller和fn.arguments获取函数调用的堆栈

增加了保留字（比如protected、static和interface）

设立"严格模式"的目的，主要有以下几个：

消除Javascript语法的一些不合理、不严谨之处，减少一些怪异行为;

消除代码运行的一些不安全之处，保证代码运行的安全；

提高编译器效率，增加运行速度；

为未来新版本的Javascript做好铺垫。

注：经过测试IE6,7,8,9均不支持严格模式。

1. defer和async区别？

script 没有 defer 和 async

会停止（阻塞）dom 树构建，立即加载，并执行脚本

script 带 async

不会停止（阻塞）dom 树构建，立即异步加载，加载好后立即执行

script 带 defer

不会停止（阻塞）dom 树构建，立即异步加载。加载好后，如果 dom 树还没构建好，则先等 dom 树解析好再执行；如果 dom 树已经准备好，则立即执行。

1. 函数防抖节流的原理（必须）

函数节流和函数防抖，两者都是优化高频率执行js代码的一种手段。

函数防抖

函数防抖是指频繁触发的情况下，只有足够的空闲时间，才执行代码一次。比如生活中的坐公交，就是一定时间内，如果有人陆续刷卡上车，司机就不会开车。只有别人没刷卡了，司机才开车。

函数防抖的应用场景，最常见的就是用户注册时候的手机号码验证和邮箱验证了。只有等用户输入完毕后，前端才需要检查格式是否正确，如果不正确，再弹出提示语。以下还是以页面元素滚动监听的例子，来进行解析：

// 函数防抖

var timer = false;

document.getElementById("debounce").onscroll = function(){

clearTimeout(timer); // 清除未执行的代码，重置回初始化状态

timer = setTimeout(function(){

console.log("函数防抖");

}, 300);

};

函数节流

函数节流是指一定时间内js方法只跑一次。比如人的眨眼睛，就是一定时间内眨一次。这是函数节流最形象的解释。

函数节流应用的实际场景，多数在监听页面元素滚动事件的时候会用到。因为滚动事件，是一个高频触发的事件。以下是监听页面元素滚动的示例代码：

// 函数节流

var canRun = true;

document.getElementById("throttle").onscroll = function(){

if(!canRun){

// 判断是否已空闲，如果在执行中，则直接return

return;

}

canRun = false;

setTimeout(function(){

console.log("函数节流");

canRun = true;

}, 300);

};

1. 请说出减少页面加载时间的方法？（必须）

尽量减少页面中重复的HTTP请求数量

服务器开启gzip压缩

css样式的定义放置在文件头部

Javascript脚本放在文件末尾

压缩合并Javascript、CSS代码

使用多域名负载网页内的多个文件、图片

1. 请介绍下缓存控制cache-control（必须）

每个资源都可以通过 Cache-Control HTTP 头来定义自己的缓存策略

Cache-Control 指令控制谁在什么条件下可以缓存响应以及可以缓存多久

Cache-Control 头在 HTTP/1.1 规范中定义，取代了之前用来定义响应缓存策略的头（例如 Expires）。

1. 一次js请求一般情况下有哪些地方会有缓存处理？（必须）

浏览器端存储

浏览器端文件缓存

HTTP缓存304

服务器端文件类型缓存

表现层&DOM缓存

1. 一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验。（重要）

图片懒加载，滚动到相应位置才加载图片。

图片预加载，如果为幻灯片、相册等，将当前展示图片的前一张和后一张优先下载。

使用CSSsprite，SVGsprite，Iconfont、Base64等技术，如果图片为css图片的话。

如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。

1. 谈谈以前端角度出发做好SEO需要考虑什么？（重要）

了解搜索引擎如何抓取网页和如何索引网页

meta标签优化

关键词分析

付费给搜索引擎

链接交换和链接广泛度（Link Popularity）

合理的标签使用

1. HTTP2.0有哪些提升？（重要）

a. HTTP 2.0中的二进制分帧层突破了限制：客户端和服务器可以把HTTP消息分解为互不依赖的  
帧，然后乱序发送，最后再在另一端把它们重新组合起来。

b. 把HTTP消息分解为很多独立的帧之后，就可以通过优化这些帧的交错和传输顺序，进一步提升  
性能。

c. HTTP 2.0通过让所有数据流共用同一个连接，可以更有效地使用TCP连接。

d. 服务器除了对最初请求的响应外，服务器还可以额外向客户端推送资源，而无需客户端明确地请  
求。

1. HTTP 2.0会压缩首部元数据，针对之前的数据只编码发送差异数据。
2. TCP与UDP的区别？（重要）

a. UDP 协议的头长度不到TCP头的一半，所以同样大小的包里UDP携带的净数据比TCP包多。

b. TCP会发响应，UDP不会。并且UDP没有Seq和Ack等概念，省去了建立连接的开销，DNS  
解析就使用UDP协议。TCP有3次握手4次挥手。

c. UDP不能分割报文段（MSS），超过MTU的时候，发送方的网络层负责分片，接收方收到分片  
后再组装起来，这个过程会消耗资源，降低性能。

1. UDP没有重传机制，丢包的时候就不能按需发送。TCP有超时重传、快速重传和SACK。
2. URI和URL的区别？

URI（Uniform Resource Identifier）：统一资源标识符。

URL（Uniform Resoure Locator）：统一资源定位符，通俗的说法是网址。

URI表示某一互联网资源，而URL表示资源地点，所以URL是URI的子集。

1. 移动端click延迟300ms和事件穿透？（重要）

解决300MS延迟

方案1：

FastClick.js插件

方案2：

用JQ的tap事件

解决事件穿透

方案1：

点击消失元素，添加超过300MS的动画

方案2：

动态地在触摸位置生成一个透明的元素，这样当上层元素消失而延迟的click来到时，它点击到的是那个透明的元素，也不会“穿透”到底下。在一定的timeout后再将生成的透明元素移除。

方案3

点击触发事件的时候pointer-events变成none,过400毫毛pointer-events变成auto

方案4

fastclick 取消 click 事件，用 touchend 模拟快速点击行为

1. JSON 对值的类型有那些要求？（重要）

原始类型的值

字符串

数值（必须以十进制表示）

布尔值

null（不能使用NaN, Infinity, -Infinity和undefined）

复合类型的值

只能是数组或对象，不能是函数、正则表达式对象、日期对象。

其他要求

字符串必须使用双引号表示，不能使用单引号。

对象的键名必须放在双引号里面。

数组或对象最后一个成员的后面，不能加逗号。

1. 对this指针的理解，可以列举几种使用情况？（必须）

纯粹的函数调用

作为对象方法的调用

作为构造函数调用

apply与call的调用

1. 解释下javascript的冒泡和捕获？

<div id="click1">

<div id="click2">

<div id="click3">事件</div>

</div>

</div>

a. Netscape主张元素1的事件首先发生，这种事件发生顺序被称为捕获型

b. 微软则保持元素3具有优先权，这种事件顺序被称为冒泡型

c. W3C选择了一个择中的方案。任何发生在w3c事件模型中的事件，首是进入捕获阶段，直到达到目标元素，再进入冒泡阶段

事件监听函数addEventListener()的第三个参数就是控制方法是捕获还是冒泡

click1 = document.getElementById("click1");

click2 = document.getElementById("click2");

click3 = document.getElementById("click3");

click1.addEventListener("click", returnTarget, false);

click2.addEventListener("click", returnTarget, false);

click3.addEventListener("click", returnTarget, false);

function returnTarget(event) {

console.log(event.currentTarget.id);

//event.stopPropagation();

}

1. ajax有哪些好处和弊端？（必须）

优点：

无刷新更新数据

异步与服务器通信

前端和后端负载平衡

基于标准被广泛支持

界面与应用分离

缺点：

AJAX干掉了Back和History功能，即对浏览器机制的破坏

AJAX的安全问题

对搜索引擎支持较弱

违背URL和资源定位的初衷

1. 简述下为何通过ajax发送的请求会出现乱码问题，如何解决？（必须）

乱码的问题就是编码格式冲突，我们需要传输中文数据前面加一个encodeURIComponent()编码，例如：encodeURIComponent( $("#keyword").val() )；在接受参数的页面对传过来的编码过后的内容用后端语言进行解码

或者使用$.ajax发送请求，设置cache参数为false

1. documen.write和 innerHTML的区别？（重要）

write

a. 直接写入到页面的内容流

b. 当在文档加载之后使用 document.write()，这会覆盖该文档，例如onload事件中。

c. 是重写整个 document, 写入内容是字符串的 html。

innerHTML

是 DOM 页面元素的一个属性，代表该元素的 html 内容，精确到某一个具体的元素来进行更改。  
将内容写入某个 DOM 节点，不会导致页面全部重绘

innerHTML 很多情况下都优于 document.write，其允许更精确的控制要刷新页面的那一个部分。

1. JavaScript 常见的内置对象？

有 Object、Math、String、Array、Number、Function、Boolean、JSON 等，其中 Object 是所有对象的基类，采用了原型继承方式。

1. JSON 的了解？（重要）

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。

它是基于 JavaScript 的一个子集。

数据格式简单，易于读写，占用带宽小。

格式：采用键值对。例如：{ “age‟: ‟12‟, ”name‟: ‟back‟ }

1. prototype 和 proto 区别是什么？（重要）

js里所有的对象都有proto属性(对象，函数)，指向构造该对象的构造函数的原型。

只有函数function才具有prototype属性。这个属性是一个指针，指向一个对象，这个对象的用途就是包含所有实例共享的属性和方法（我们把这个对象叫做原型对象）。原型对象也有一个属性，叫做constructor，这个属性包含了一个指针，指回原构造函数。

1. 什么是变量提升?

变量提升是面试中非常常见的一个问题

比如 var 和 let 的区别是什么？

答案之一就有 let 不会变量提升

1. \*\* 和 % 分别是什么运算符？

\*\* 指数运算 % 取余运算

1. js中改变原数组的方法以及不改变原数组的方法？（必须）

改变原数组

push方法用于在数组的末端添加一个或多个元素，并返回添加新元素后的数组长度。

pop方法用于删除数组的最后一个元素，并返回该元素。

shift方法用于删除数组的第一个元素，并返回该元素。

unshift方法用于在数组的第一个位置添加元素，并返回添加新元素后的数组长度。注意，该方法会改变原数组。

reverse方法用于颠倒排列数组元素，返回改变后的数组。

splice方法用于删除原数组的一部分成员，并可以在删除的位置添加新的数组成员，返回值是被删除的元素。

sort方法对数组成员进行排序，默认是按照字典顺序排序。

不改变原数组

concat方法用于多个数组的合并。它将新数组的成员，添加到原数组成员的后部，然后返回一个新数组。

slice方法用于提取目标数组的一部分，返回一个新数组。

map方法将数组的所有成员依次传入参数函数，然后把每一次的执行结果组成一个新数组返回。

filter方法用于过滤数组成员，满足条件的成员组成一个新数组返回。

1. 介绍下 promise.all? （必须）

Promise.all()方法将多个Promise实例包装成一个Promise对象（p），接受一个数组（p1,p2,p3）作为参数，数组中不一定需要都是Promise对象，但是一定具有Iterator接口，如果不是的话，就会调用Promise.resolve将其转化为Promise对象之后再进行处理。

使用Promise.all()生成的Promise对象（p）的状态是由数组中的Promise对象（p1,p2,p3）决定的。

1. 如果所有的Promise对象（p1,p2,p3）都变成fullfilled状态的话，生成的Promise对象（p）  
也会变成fullfilled状态，p1,p2,p3三个Promise对象产生的结果会组成一个数组返回给传递给p的回调函数。

2. 如果p1,p2,p3中有一个Promise对象变为rejected状态的话，p也会变成rejected状态，第一  
个被rejected的对象的返回值会传递给p的回调函数。

Promise.all()方法生成的Promise对象也会有一个catch方法来捕获错误处理，但是如果数组中的

Promise对象变成rejected状态时，并且这个对象还定义了catch的方法，那么rejected的对象会执行自己的catch方法。

并且返回一个状态为fullfilled的Promise对象，Promise.all()生成的对象会接受这个Promise对象，

不会返回rejected状态。

1. 函数call、apply和bind的区别？（必须）

call、apply、bind都是改变this指向的方法

call和apply第2个参数不同，call是函数参数序列，而apply是数组形式

bind返回对应函数, 便于稍后调用，apply和 call则是立即调用。

在 ES6 的箭头函数下, call 和 apply 将失效

1. 什么是原型对象？（重要）

原型对象本质上是一个普通对象

js原型对象的引入为了解决继承问题

每个函数都有一个prototype属性，指向一个对象

当实例对象本身没有某个属性或方法的时候，它会到原型对象去寻找该属性或方法。

如果实例对象自身就有某个属性或方法，它就不会再去原型对象寻找这个属性或方法。

1. 什么是原型链？（重要）

1. 每一个对象都一个原型对象prototype

2.由于原型对象也是对象，所以它也有自己的原型对象\_\_proto\_\_

3. 因此，就会形成一个“原型链”

4. 对象到原型,再到原型的原型

1. 什么是constructor 属性？（重要）

原型prototype对象有一个constructor属性，默认指向prototype对象所在的构造函数。

由于constructor属性定义在prototype对象上面，意味着可以被所有实例对象继承。

1. 什么是instanceof 关键字？（重要）

instanceof运算符返回一个布尔值，表示对象是否为某个构造函数的实例

1. typeof和instanceof都可以用来判断变量？

typeof用以获取一个变量或者表达式的类型，typeof一般只能返回如下几个结果：

number,boolean,string,function（函数）,object（NULL,数组，对象）, undefined。

如：

alert(typeof (123));//typeof(123)返回"number"

alert(typeof ("123"));//typeof("123")返回"string"

我们可以使用typeof来获取一个变量是否存在，如if(typeof a!="undefined"){}，而不要去使用if(a)因为如果a不存在（未声明）则会出错，正因为typeof遇到null,数组,对象时都会返回object类型，所以当我们要判断一个对象是否是数组时或者判断某个变量是否是某个对象的实例则要选择使用另一个关键语法instanceof

instanceof用于判断一个变量是否某个对象的实例，如

var a=new Array();

alert(a instanceof Array); //会返回true，

同时alert(a instanceof Object)也会返回true; 这是因为Array是object的子类。

再如：

function test(){};

var a=new test();

alert(a instanceof test) //会返回true。

1. 栈和堆的区别？（重要）

栈（stack）

由编译器自动分配释放，存放函数的参数值，局部变量等；

堆（heap）

一般由程序员分配释放，若程序员不释放，程序结束时可能由操作系统释放。

1. 什么是宏任务和微任务？（重要）

宏任务

包括整体代码 script

按指定间隔执行1次 setTimeout

按指定间隔重复执行 setInterval

UI交互事件

微任务

Promise.then

Object.observe

process.nextTick(Node.js下)

1. 线程与进程的区别？（重要）

线程：是程序执行流的最小单元，是系统独立调度和分配[CPU](http://m.elecfans.com/article/645277.html" \t "http://www.elecfans.com/d/_blank)（独立运行）的基本单位。

进程：是资源分配的基本单位。一个进程包括多个线程。

两者区别

1、线程与资源分配无关，它属于某一个进程，并与进程内的其他线程一起共享进程的资源。

2、每个进程都有自己一套独立的资源（数据），供其内的所有线程共享。

3、不论是大小，开销线程要更“轻量级”

4、一个进程内的线程通信比进程之间的通信更快速，有效。（因为共享变量）

1. 如何理解js是单线程模型？（重要）

js 只在一个线程上运行

js 同时只能执行一个任务，其他任务都必须在后面排队等待

js 引擎可以有多个线程，单个脚本只能在一个主线程上运行

1. 介绍一下async 和 await （重要）

主要考察宏任务和微任务，搭配promise，询问一些输出的顺序

原理：async 和 await 用了同步的方式去做异步，async 定义的函数的返回值都是 promise，await

后面的函数会先执行一遍，然后就会跳出整个 async 函数来执行后面js栈的代码

1. ES6 的 class 和构造函数的区别? （重要）

class 的写法只是语法糖，和之前 prototype 差不多，但还是有细微差别的，下面看看：

1. 严格模式

类和模块的内部，默认就是严格模式，所以不需要使用 use strict 指定运行模式。只要你的代码写在类或模块之中，就只有严格模式可用。考虑到未来所有的代码，其实都是运行在模块之中，所以 ES6 实际上把整个语言升级到了严格模式。

2. 不存在提升

类不存在变量提升（hoist），这一点与 ES5 完全不同。

复制new Foo(); // ReferenceError class Foo {}

3. 方法默认是不可枚举的

ES6 中的 class，它的方法（包括静态方法和实例方法）默认是不可枚举的，而构造函数默认是可枚举的。细想一下，这其实是个优化，让你在遍历时候，不需要再判断 hasOwnProperty 了

4. class 的所有方法（包括静态方法和实例方法）都没有原型对象 prototype，所以也没有

[[construct]]，不能使用 new 来调用。

5. class 必须使用 new 调用，否则会报错。这是它跟普通构造函数的一个主要区别，后者不用 new 也可以执行。

6. ES5 和 ES6 子类 this 生成顺序不同

ES5 的继承先生成了子类实例，再调用父类的构造函数修饰子类实例。ES6 的继承先 生成父类实例，再调用子类的构造函数修饰父类实例。这个差别使得 ES6 可以继承内置对象。

1. ES6可以继承静态方法，而构造函数不能
2. polyfill 是什么？（重要）

Polyfill你可以理解为“腻子”，就是装修的时候，可以把缺损的地方填充抹平。

举个例子，html5的storage(session,local), 不同浏览器，不同版本，有些支持，有些不支持。

我们又想使用这个特性，怎么办？有些人就写对应的Polyfill（Polyfill有很多），帮你把这些差异化抹平，不支持的变得支持了

1. ES6 声明变量的六种方法? （重要）

var命令

function命令

let命令

const命令

import命令

class命令

1. es6新特性有哪些？（必须）

let 和 const 块级作用域

数组和对象解构赋值

扩展运算符

Set 和 Map 数据结构

import和export 模块化加载

箭头函数

Proxy 对象

Promise 对象

Iterator 迭代器对象

Generator 函数

async 函数

Class 类的实现

1. super关键字有什么用？（重要）

super不仅仅是一个关键字，还可以作为函数和对象

作为函数

在子类继承父类中，super作为函数调用，只能写在子类的构造函数（constructor）里面，并且在this调用之前，super代表的是父类的构造函数。

作为对象

在普通方法中使用，super指向父类的原型

用在静态方法中使用，super将指向父类

1. 箭头函数有几个使用注意点？（必须）
2. 函数体内的this对象，就是定义时所在的对象，而不是使用时所在的对象
3. 不可以当作构造函数，也就是说，不可以使用new命令，否则会抛出一个错误
4. 不可以使用arguments对象，该对象在函数体内不存在。如果要用，可以用 rest 参数代替。
5. 不可以使用yield命令，因此箭头函数不能用作 Generator 函数。
6. 介绍一下Set和Map的区别? （必须）

Set本身就是一个构造函数常用于数据重组。Map类似于对象，也是键值对的集合，但是Map的键不仅仅只限于字符串，各种类型的值（包括对象）都可以当作键。

Set的特点

（1）成员不能重复

（2）只有键值没有键名，类似数组

（3）可以遍历，方法有add, delete,has

Map的特点

（1）本质上是健值对的集合，类似集合

（2）可以遍历，可以跟各种数据格式转换

1. Iterator是什么，有什么作用？

Iterator是ES6中一个很重要概念，它并不是对象，也不是任何一种数据类型。因为ES6新增了Set、Map类型，他们和Array、Object类型很像，Array、Object都是可以遍历的，但是Set、Map都不能用for循环遍历，解决这个问题有两种方案，一种是为Set、Map单独新增一个用来遍历的API，另一种是为Set、Map、Array、Object新增一个统一的遍历API，显然，第二种更好，ES6也就顺其自然的需要一种设计标准，来统一所有可遍历类型的遍历方式。Iterator正是这样一种标准。或者说是一种规范理念

1. Generator函数是什么，有什么作用？

如果说JavaScript是ECMAScript标准的一种具体实现、Iterator遍历器是Iterator的具体实现，那么Generator函数可以说是Iterator接口的具体实现方式。

执行Generator函数会返回一个遍历器对象，每一次Generator函数里面的yield都相当一次遍历器对象的next()方法，并且可以通过next(value)方法传入自定义的value,来改变Generator函数的行为。

Generator函数可以通过配合Thunk 函数更轻松更优雅的实现异步编程和控制流管理。

1. async函数是什么，有什么作用？（重要）

async函数可以理解为内置自动执行器的Generator函数语法糖，它配合ES6的Promise近乎完美的实现了异步编程解决方案

1. Class、extends是什么，有什么作用？（重要）

ES6 的class可以看作只是一个ES5生成实例对象的构造函数的语法糖。它参考了java语言，定义了一个类的概念，让对象原型写法更加清晰，对象实例化更像是一种面向对象编程。Class类可以通过extends实现继承。它和ES5构造函数的不同点

1. resolve 和 reject是什么参数？（重要）

参数resolve和reject的作用是将Promise中函数要传递的值，作为参数传给后面的then和catch中函数。

1. Promise.all、Promise.race和Promise.allSettled的区别？（重要）

Promise.all()

Promise.all() 方法返回一个 Promise 实例，此实例在参数内所有的 promise实例都“完成（resolved）”或参数中不包含 promise实例时回调完成（resolve）；如果参数中 promise实例 有一个失败（rejected），此实例回调失败（reject），失败原因的是第一个失败 promise 的结果。

Promise.race()

Promise.race(iterable) 方法返回一个 promise实例，一旦迭代器中的某个promise实例解决（完成）或拒绝（失败），返回的 promise实例就会解决或拒绝。

Promise.allSettled()

Promise.allSettled()方法返回一个promise实例，该promise实例在所有给定的promise已被解析或被拒绝后解析，并且每个对象都描述每个promise实例的结果。

1. module、export、import是什么，有什么作用？（重要）

module、export、import是ES6用来统一前端模块化方案的设计思路和实现方案。export、import的出现统一了前端模块化的实现方案，整合规范了浏览器/服务端的模块化方法，用来取代传统的AMD/CMD、requireJS、seaJS、commondJS等等一系列前端模块不同的实现方案，使前端模块化更加统一规范，JS也能更加能实现大型的应用程序开发。

import引入的模块是静态加载（编译阶段加载）而不是动态加载（运行时加载）。

import引入export导出的接口值是动态绑定关系，即通过该接口，可以取到模块内部实时的值

1. 说说你对Promise的理解？（必须）

依照 Promise/A+ 的定义，Promise 有四种状态：

pending: 初始状态, 非 fulfilled 或 rejected.

fulfilled: 成功的操作.

rejected: 失败的操作.

settled: Promise已被fulfilled或rejected，且不是pending

另外， fulfilled 与 rejected 一起合称 settled

Promise 对象用来进行延迟(deferred) 和异步(asynchronous ) 计算

1. 介绍一下模板字符串有哪些特点？
2. 传统的输出模板需要使用单引号搭配转义符拼接实现多行输出，这种方式写法繁琐且不方便，es6  
    引入反引号作为标识符可以在 `` 内部直接换行实现多行输出。
3. 模板字符串中嵌入变量，需要将变量名写在${}之中，调用函数同样写在${}中。
4. 使用模板字符串表示多行字符串，所有的空格和缩进都会被保留在输出之中。
5. 串中使用反引号`需要使用转义字符转义
6. 什么是Symbol?（必须）

Symbol是ES6 引入了一种新的原始数据类型Symbol，表示独一无二的值它是 JavaScript 语言的第七种数据类型。

Symbol的 值通过Symbol函数生成，凡是属于 Symbol类型的属性都是独一无二的，Symbol类型可以保证不会与其他属性名产生冲突。

Symbol值不能与其他类型的值进行运算，会报错。

Symbol值也可以转为布尔值，但是不能转为数值。

1. 介绍一下解构赋值? （重要）

概念：从数组和对象中提取值，对变量进行赋值

数组解构：元素是按次序排列的，变量的取值由它的位置决定，如：

let [a, b, c] = [1, 2, 3];

对象结构：对象的属性没有次序，变量必须与属性同名，才能取到正确的值，如

let { bar, foo } = { foo: 'aaa', bar: 'bbb' };

foo // "aaa"

bar // "bbb"

1. js中==、===和Object.is()的区别？（重要）

==：等同，比较运算符，两边值类型不同的时候，先进行类型转换，再比较。

===：恒等，严格比较运算符，不做类型转换，类型不同就是不等。

Object.is(): 是ES6新增的用来比较两个值是否严格相等的方法，与===的行为基本一致，区别是Object.is()函数支持 +0 不等于 -0和NaN等于NaN。

1. ECMAScript 6 怎么写 class ，为何会出现class？

ES6的class可以看作是一个语法糖，它的绝大部分功能ES5都可以做到，新的class写法只是让对象原型的写法更加清晰、更像面向对象编程的语法.

//定义类

class Point {

constructor(x,y) {

//构造方法

this.x = x; //this关键字代表实例对象

this.y = y;

} toString() {

return '(' + this.x + ',' + this.y + ')';

}

}

1. ES6实现继承的方式？（必须）

ES6中通过extends关键字实现继承，案例如下

class Father {

constructor（x，y）{

this.x = x;

this.y = y

}

sum (){

console.log(this.x + this.y)

}

}

class Son extends Father{

constructor(x,y){

super(x,y)

}

}

//子类继承父类，调用父类中的函数时，需要使用super关键字将值传给父类  
//这样才能调用父类中的方法

var son = new Son(1,2)

son.sum()//3

1. 谈谈你对Promise的理解？（必须）

Promise引入有效的解决了异步任务回调地狱问题，传统的异步任务使用回调函数来实现，当多个异步任务直接存在一定执行顺序时，后面的异步任务需要在前面的任务完成之后在执行，后面的异步任务需要现在前面一个任务执行结束的回调函数中来执行，这样就会形成回调函数一层套一层的现象，这样多个异步任务之间耦合性就相对较高，为了解决这一问题，引入了promise，promise以同步的书写方式实现了异步的处理，Promise通过promise对象.then的方式实现传统异步任务回调函数的处理，以及下一个异步任务的执行。使用promise需要返回一个promise对象才能实现下一次.then的调用。

1. Promise构造函数是同步执行还是异步执行，那么 then 方法呢？（必须）

promise构造函数是同步执行的，then方法是异步执行的

1. promise有几种状态，什么时候会进入catch？（必须）

三个状态

pending、resolve、reject

两个过程

padding -> resolve、padding -> rejected当pending为rejectd时，会进入catch

1. setTimeout、Promise、Async/Await 的区别? （重要）

事件循环中分为宏任务队列和微任务队列

其中setTimeout的回调函数放到宏任务队列里，等到执行栈清空以后执行promise.then里的回调函数会放到相应宏任务的微任务队列里，等宏任务里面的同步代码执行完再执行async函数表示函数里面可能会有异步方法，await后面跟一个表达式 async方法执行时，遇到await会立即执行表达式，然后把表达式后面的代码放到微任务队列里，让出执行栈让同步代码先执行

1. object.defineproporty与proxy的区别?（重要）

proxy更加的方便，它下面有个handle有许多的操作方法。

proxy是可以监听到数组的变化

proxy还可以监听到添加新属性的

proxy监听的是整个对象，object.defineproporty只是监听了对象下面的一个属性。

1. 介绍一下你了解的异步任务处理方式？ （重要）

使用回调函数的形式处理异步任务，这种方式会形成回调地狱问题。

使用promise通过在then方法中处理上一次的执行结果以及下一次异步任务的发起，这种方式不会形成回调地狱问题。

使用Generator实现异步，通过yield表达式来实现异步任务的发起，以及自动执行函数，实现异步任务的结果处理。

基于async/await来实现异步任务的处理，使用async来声明任务执行函数，在函数中通过await来发起异步任务。

1. Promise中reject 和 catch处理上有什么区别? （重要）

reject 是用来抛出异常，catch是用来处理异常

reject 是 Promise 的方法，而 catch 是 Promise 实例的方法

reject后的东西，一定会进入then中的第二个回调，如果then中没有写第二个回调，则进入catch

网络异常（比如断网），会直接进入catch而不会进入then的第二个回调

1. rest(剩余参数)和arguments对象的区别是什么? （重要）
2. 剩余参数只包含那些没有对应形参的实参(可以是参数的一部分)，而arguments对象包含了传给函数的所有实参(是参数的全部)。

(2) arguments 对象不是一个真实的数组，而剩余参数是真实的Array 实例。也就是说，能够在它上面直接使用所有的数组方法，比如sort、map、forEach、 pop。

(3) arguments对象还有一些附加的属性(如callee属性)。

1. 如果想在arguments对象上使用数组方法，首先要将它转换为真实的数组，比如使用[].slic.call(arguments)。
2. async/await 以及对于Generator的优势？（重要）

async await 是用来解决异步的，async 函数是 Generator 函数的语法糖

使用关键字 async 来表示，在函数内部使用await来表 示异步.

async 函数返回一个 Promise 对象，可以使用 then 方 法添加回调函数

当函数执行的时候，一旦遇到 await 就会先返回，等到 异步操作完成，再接着执行函数体内后面的语句。

1. forEach 、for in 、for of 三者区别? （重要）

forEach 更多的用来遍历数组

for in 一般常用来遍历对象或 json

for of 数组对象都可以遍历，遍历对象需要通过和Object.keys()

for in 循环出的是 key，for of 循环出的是 value

1. 箭头函数与普通函数有什么区别？（重要）

相比普通函数更简洁的语法

不能使用new

不绑定arguments，用rest参数...解决

捕获其所在上下文的 this 值，作为自己的 this 值

箭头函数没有原型属性

不能简单返回对象字面量

箭头函数不能当做Generator函数,不能使用yield关键字

箭头函数不能换行

1. 举一些ES6对String字符串类型做的常用升级优化？（重要）

优化部分

字符串模板

升级部分

字符串原型上新增

includes() 查找字符串中是否包含某字符，返回布尔值。

startsWith() 字符串是否以某字符开头，返回布尔值。

endsWith() 字符串是否以某字符结尾，返回布尔值。

padStart() 在字符串开头补全指定长度的字符。

padEnd() 在字符串结尾补全指定长度的字符。

repeat() 重复指定次数字符。

1. 举一些ES6对Array数组类型做的常用升级优化？（重要）

优化部分

数组解构赋值

扩展运算符

升级部分

find() 方法

indexOf() 查找数组指定成员的索引值 （修复了indexOf查找不到NaN的bug）

includes() 查找数组是否包含指定的成语，返回布尔值

fill() 使用给定值，填充一个数组

flat() 推平数组

1. 举一些ES6对Number数字类型做的常用升级优化？（重要）

优化部分

ES6在Number原型上新增了isFinite(), isNaN()方法，用来取代传统的全局isFinite(), isNaN()方法检测数值是否有限、是否是NaN。ES5的isFinite()和isNaN()方法都会先将非数值类型的参数转化为Number类型再做判断，这其实是不合理的，最造成isNaN('NaN') === true的奇怪行为'NaN'是一个字符串，但是isNaN却说这就是NaN。而Number.isFinite()和Number.isNaN()则不会有此类问题(Number.isNaN('NaN') === false)。（isFinite()同上）

升级部分

ES6在Math对象上新增了Math.cbrt()，trunc()，hypot()等等较多的科学计数法运算方法，可以更加全面的进行立方根、求和立方根等等科学计算

1. 举一些ES6对Object类型做的常用升级优化? （重要）

优化部分

对象解构赋值

对象属性名缩写

对象属性方法缩写

super关键字总是指向原型对象

升级部分

Object.is() 比较目标对象的值，支持NaN相等

Object.assign() 用于对象新增属性或者多个对象合并

对象实例上新增

getOwnPropertyDescriptors() 返回对象自身属性的对象描述信息

getPrototypeOf() 和setPrototypeOf() 方法，用来获取或设置当前对象的prototype对象

Object.keys()、Object.values() 和 Object.entries()

1. 举一些ES6对Function函数类型做的常用升级优化? （重要）

优化部分

箭头函数

函数默认赋值

rest参数取代arguments

升级部分

ES6新增了双冒号运算符，用来取代以往的bind，call,和apply。(浏览器暂不支持，Babel已经支持转码)

foo::bar; // 等同于下面的代码

bar.bind(foo);

1. 什么是模块化开发?

模块就是一个有特定功能的文件，我们可以通过加载这些模块得到特定的功能

模块化开发就是js的功能分离，通过需求引入不同的文件

模块化开发可以使代码耦合度降低，避免代码多次在页面出现,他最大的作用就是重用

模块开发要遵循的规范

AMD规范也叫异步模块加载规范,在这个规范下模块会异步加载,不影响后面语句的执行,我们可以使用define定义模块,使用require调用模块

CommonJS规范是服务器端模块的规范,node.js就采用了这个规范,每个模块都有一个单独的作用域,模块内部的变量无法被其他模块读取,除非定义为global的对象和属性

CMD规范通用模块定义.CMD是按需加载,一个模块就是一个文件

1. forEach、for in、for of三者区别？ （重要）

forEach更多的用来遍历数组

for in 一般常用来遍历对象或json

for of数组对象都可以遍历，遍历对象需要通过和Object.keys()

for in循环出的是key，for of循环出的是value

1. 谈谈你对AMD、CMD的理解？

一、CommonJS是服务器端模块的规范，Node.js采用了这个规范。CommonJS规范加载模块是同步的，也就是说，只有加载完成，才能执行后面的操作。AMD规范则是非同步加载模块，允许指定回调函数。

二、AMD推荐的风格通过返回一个对象做为模块对象，CommonJS的风格通过对module.exports或exports的属性赋值来达到暴露模块对象的目的。

三、CommonJS 的规范中，每个 JavaScript 文件就是一个独立的模块上下文（module context），在这个上下文中默认创建的属性都是私有的。也就是说，在一个文件定义的变量（还包括函数和类），都是私有的，对其他文件是不可见的。

四、CommonJS是同步加载模块,在浏览器中会出现堵塞情况，所以不适用。

五、AMD 异步，需要定义回调define方式

六、es6 一个模块就是一个独立的文件，该文件内部的所有变量，外部无法获取。如果你希望外部能够读取模块内部的某个变量，就必须使用export关键字输出该变量 es6还可以导出类、方法，自动适用严格模式。

1. 什么是webpack？（重要）

webpack 是一个模块打包工具，在 webpack 里面一切皆模块通过 loader转换文件，通过plugin注入钩子，最后输出有多个模块组合成的文件，WebPack可以看做是模块打包机：它做的事情是，分析你的项目结构，找到Js模块以及其它的一些浏览器不能直接运行的拓展语言，并将其打包为合适的格式以供浏览器使用。

1. webpack的优缺点? （重要）

优点

专注于处理模块化的项目，能做到开箱即用，一步到位。

可通过plugin扩展，方便、灵活。

使用场景不局限于web开发。

社区庞大活跃，经常引入紧跟时代发展的新特性，能为大多数场景找到已有的开源扩展。

良好的开发体验。

缺点

只能用于采用模块化开发的项目

1. webpack与grunt、gulp的不同？

三者都是前端构建工具，grunt和gulp在早期比较流行，现在webpack相对来说比较主流,不过一些轻量化的任务还是会用gulp来处理，比如单独打包CSS文件等。

grunt和gulp是基于任务和流(Task、Stream)的。类似jQuery，找到一个（或一类）文件，对其做一系列链式操作，更新流上的数据， 整条链式操作构成了一个任务，多个任务就构成了整个web的构建流程。

webpack是基于入口的。webpack会自动地递归解析入口所需要加载的所有资源文件，然后用不同的Loader来处理不同的文件，用Plugin来扩展webpack功能。

所以总结一下：

从构建思路来说

gulp和grunt需要开发者将整个前端构建过程拆分成多个Task，并合理控制所有Task的调用关系 webpack需要开发者找到入口，并需要清楚对于不同的资源应该使用什么Loader做何种解析和加工

对于知识背景来说

gulp更像后端开发者的思路，需要对于整个流程了如指掌 webpack更倾向于前端开发者的思路

1. webpack的构建流程是什么? （重要）

Webpack 的运行流程是一个串行的过程，从启动到结束会依次执行以下流程：

初始化参数：从配置文件和 Shell 语句中读取与合并参数，得出最终的参数；

开始编译：用上一步得到的参数初始化 Compiler 对象，加载所有配置的插件，执行对象的 run 方法开始执行编译；

确定入口：根据配置中的 entry 找出所有的入口文件；

编译模块：从入口文件出发，调用所有配置的 Loader 对模块进行翻译，再找出该模块依赖的模块，再递归本步骤直到所有入口依赖的文件都经过了本步骤的处理；

完成模块编译：在经过第4步使用 Loader 翻译完所有模块后，得到了每个模块被翻译后的最终内容以及它们之间的依赖关系；

输出资源：根据入口和模块之间的依赖关系，组装成一个个包含多个模块的 Chunk，再把每个 Chunk 转换成一个单独的文件加入到输出列表，这步是可以修改输出内容的最后机会；

输出完成：在确定好输出内容后，根据配置确定输出的路径和文件名，把文件内容写入到文件系统。

在以上过程中，Webpack 会在特定的时间点广播出特定的事件，插件在监听到感兴趣的事件后会执行特定的逻辑，并且插件可以调用 Webpack 提供的 API 改变 Webpack 的运行结果。

1. 有哪些常见的Loader？他们是解决什么问题的 （重要）

file-loader：把文件输出到一个文件夹中，在代码中通过相对 URL 去引用输出的文件 url-loader：和 file-loader 类似，但是能在文件很小的情况下以 base64 的方式把文件内容注入到代码中去 source-map-loader：加载额外的 Source Map文件，以方便断点调试 image-loader：加载并且压缩图片文件 babel-loader：把 ES6 转换成 ES5 css-loader：加载 CSS，支持模块化、压缩、文件导入等特性 style-loader：把 CSS 代码注入到 JavaScript 中，通过 DOM 操作去加载 CSS。 eslint-loader：通过 ESLint 检查 JavaScript 代码

1. babel是什么，有什么作用?

babel是一个 ES6 转码器，可以将 ES6 代码转为 ES5 代码，以便兼容那些还没支持ES6的平台

1. let有什么用，有了var为什么还要用let？

在ES6之前，声明变量只能用var，var方式声明变量其实是很不合理的，准确的说，是因为ES5里面没有块级作用域是很不合理的。没有块级作用域回来带很多难以理解的问题，比如for循环var变量泄露，变量覆盖等问题。let声明的变量拥有自己的块级作用域，且修复了var声明变量带来的变量提升问题。

1. 有哪些常见的Plugin？他们是解决什么问题的？

define-plugin：定义环境变量 commons-chunk-plugin：提取公共代码 uglifyjs-webpack-plugin：通过UglifyES压缩ES6代码

1. 如何利用webpack来优化前端性能？（提高性能和体验）

用webpack优化前端性能是指优化webpack的输出结果，让打包的最终结果在浏览器运行快速高效。

压缩代码。删除多余的代码、注释、简化代码的写法等等方式。可以利用webpack的UglifyJsPlugin和ParallelUglifyPlugin来压缩JS文件， 利用cssnano（css-loader?minimize）来压缩css

利用CDN加速。在构建过程中，将引用的静态资源路径修改为CDN上对应的路径。可以利用webpack对于output参数和各loader的publicPath参数来修改资源路径

删除死代码（Tree Shaking）。将代码中永远不会走到的片段删除掉。可以通过在启动webpack时追加参数--optimize-minimize来实现

提取公共代码。

1. 如何提高webpack的构建速度

多入口情况下，使用CommonsChunkPlugin来提取公共代码

通过externals配置来提取常用库

利用DllPlugin和DllReferencePlugin预编译资源模块 通过DllPlugin来对那些我们引用但是绝对不会修改的npm包来进行预编译，再通过DllReferencePlugin将预编译的模块加载进来。

使用Happypack 实现多线程加速编译

使用webpack-uglify-parallel来提升uglifyPlugin的压缩速度。 原理上webpack-uglify-parallel采用了多核并行压缩来提升压缩速度

使用Tree-shaking和Scope Hoisting来剔除多余代码

1. 什么是错误优先的回调函数？

错误优先的回调函数用于传递错误和数据。第一个参数始终应该是一个错误对象， 用于检查程序是否发生了错误，其余的参数用于传递数据。

1. 如何避免回调地狱？（必须）

你可以有如下几个方法：

1.模块化：将回调函数分割为独立的函数

2.使用Promises

3.使用yield

4.来计算生成器或Promise

解析：这个问题有很多种答案，取决你使用的场景，例如ES6, ES7，或者一些控制流库。

1. 什么是事件循环？（必须）
2. Node采用的是单线程的处理机制（所有的I/O请求都采用非阻塞的工作方式），至少从Node.js  
    开发者的角度是这样的。 而在底层，Node.js借助libuv来作为抽象封装层， 从而屏蔽不同操  
    作系统的差异，Node可以借助livuv来来实现多线程。
3. Libuv库负责Node API的执行。它将不同的任务分配给不同的线程，形成一个事件循环， 以异  
    步的方式将任务的执行结果返回给V8引擎。
4. 每一个I/O都需要一个回调函数——一旦执行完便推到事件循环上用于执行。

解释：这用于检查Node.js的底层知识，例如什么是libuv，它的作用是什么。

1. 如何用Node监听80端口？

这题有陷阱！在类Unix系统中你不应该尝试去监听80端口，因为这需要超级用户权限。 因此不推荐让你的应用直接监听这个端口。

目前，如果你一定要让你的应用监听80端口的话，你可以有通过在Node应用的前方再增加一层反向代理 （例如nginx）来实现。否则，建议你直接监听大于1024的端口。

解释：这个问题用于检查被面试者是否有实际运行Node应用的经验。

1. 哪些工具可以用来保证一致性的代码风格？（重要）

你可以选择如下的工具

JSLint

JSHint

ESLint

JSCS - 推荐

在团队开发中，这些工具对于编写代码非常的有帮助，能够帮助团队开发者强制执行规定的风格指南， 还能够通过静态分析捕获常见的错误。

解析：用于检查被面试者是否有大型项目开发经验。

1. 使用NPM有哪些好处？

通过NPM，你可以安装和管理项目的依赖，并且能够指明依赖项的具体版本号。 对于Node应用开发而言，你可以通过package.json文件来管理项目信息，配置脚本， 以及指明项目依赖的具体版本。

解析：它能考察面试者使用npm命令的基础知识和Node.js开发的实际经验。

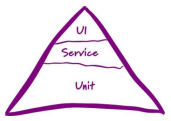
1. 什么是Stub？举个使用场景？

Stub是用于模拟一个组件或模块的函数或程序。在测试用例中， 简单的说，你可以用Stub去模拟一个方法，从而避免调用真实的方法， 使用Stub你还可以返回虚构的结果。你可以配合断言使用Stub。

解析：用于测试被面试者是否有测试的经验。如果被面试者知道什么是Stub， 那么可以继续问他是如何做单元测试的。

1. 什么是测试金字塔？

测试金字塔指的是： 当我们在编写测试用例时，底层的单元测试应该远比上层的端到端测试要多。



当我们谈到HTTP API时，我们可能会涉及到：

有很多针对模型的底层单元测试

但你需要测试模型间如何交互时，需要减少集成测试

1. 浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢？（重要）

在线的情况下，浏览器发现html头部有manifest属性，它会请求manifest文件，如果是第一次访问app，那么浏览器就会根据manifest文件的内容下载相应的资源并且进行离线存储。如果已经访问过app并且资源已经离线存储了，那么浏览器就会使用离线的资源加载页面，然后浏览器会对比新的manifest文件与旧的manifest文件，如果文件没有发生改变，就不做任何操作，如果文件改变了，那么就会重新下载文件中的资源并进行离线存储。

离线的情况下，浏览器就直接使用离线存储的资源

jQuery专题

1. 什么是jQuery？（重要）

jQuery是一个快速，轻量且功能丰富的客户端JavaScript库/框架， 它有助于遍历HTML DOM，制作动画，添加Ajax交互，操纵页面内容， 更改样式并提供很酷的UI效果。 它是最受欢迎的客户端库之一，根据调查，它在每个第二个网站上就会有一个使用。

1. 为什么我们使用jQuery？（重要）

易于使用和学习。 易于扩展。 跨浏览器支持（IE 6.0 +，FF 1.5 +，Safari 2.0 +，Opera 9.0+） 易于用于DOM操作和遍历。 大量内置方法。 AJAX功能。 更改或应用CSS，创建动画的方法。 事件检测和处理。 满足各种需求的大量插件。

1. jQuery和javaScript有什么区别？（重要）

JavaScript 是一种脚本语言，主要用在浏览器中，实现对网页的文档对象的操作和一些用户交互动作的处理。

而 jQuery 则是 JavaScript 的一个代码库（或习惯性叫类库），它将一些在 JavaScript 开发中经常用到的功能集合起来，以方便开发者直接使用，而不需要再用原生 JavaScript 语句写大量代码，同时可在不同浏览器间实现一致的效果，jQuery的宗旨是写的少、做的多，是当前最流行的 JavaScript 库之一。

1. 库和框架的区别？

库相当于工具箱，里面包含了各种工具，框架就是提供一套解决方案，拿过来就可以用，开箱即用。框架里面可以包含js库，框架比如有前端框bootstrap、elementUI、antd和vant等，里面包含了各种组件解决方案。

1. jquery的优点？（必须）

轻量级

强大的选择器

出色的DOM操作封装

可靠的事件处理机制

出色的浏览器兼容性

完善的ajax支持

封装了js动画支持

具有特色的链式操作

插件扩展开发

1. CDN有什么优缺点？（重要）

CDN 内容分发技术，它是指将资源分配到各大主干机房节点上，然后访客可以就近拉取数据，这样可以节省访问时间。

优点

本地缓存加速，提高了企业站点的访问速度

跨运营商的网络加速，保证不同网络的用户都得到良好的访问质量

远程用户访问时从缓存服务器上读取数据，减少远程访问的带宽、分担网络流量、减轻原站点web服务器负载等功能

广泛分布的CDN节点加上节点之间的智能冗余机制，可以有效地预防黑客入侵。

缺点

运营成本很高

一些国家会限制不能访问你的网站

项目的源码等失去控制

1. jQuery 库中的 $() 是什么？（必须）

$是jQuery的别称，它一个工厂函数，这个工厂函数的参数，可以放选择器字符串，也可以放一个js原生对象。它的作用是将目标对象包装为jquery对象，这样就可以使用jquery的各种方法，甚至可以使用链式操作。

1. 什么是 jQuery 对象？

jquery对象是一个类数组对象，里面的成员是匹配的原生dom对象，它有length长度，也可以通过索引值取出原生dom对象

1. 原生对象和jquery对象如何互转？（重要）

原生对象转jquery对象

var box = document.getElementById(‘box’);  
var $box = $(box)

通过$工厂函数可以转换为jquery对象

jquery对象转原生对象

var $box = $(‘#box’);  
var box = $box[0];  
var box = $box.get(0);

通过数组索引值或者通过get函数转换为原生对象

1. this 和 $(this) 有什么区别？（必须）

$(this) 使用了$工厂函数包装，所以它是一个jquery对象，而this则是js原生对象。

1. $(document).ready()与window.onload的区别？（必须）

执行时间不同

window.onload必须等到页面内包括图片的所有元素加载完毕后才能执行。

$(document).ready()是DOM结构绘制完毕后就执行，不必等到加载完毕。

编写个数不同

window.onload不能同时编写多个，如果有多个window.onload方法，只会执行一个，也就是后面的会覆盖前面的事件。

简化写法

window.onload 没有简化写法

$(document).ready(function(){})可以简写成$(function(){});

1. 使用 CDN 加载 jQuery 库的主要优势是什么 ?（必须）

CDN的主要目标是为最终用户提供高可用性和高性能的内容。

使用CDN的优点：

它减少了服务器的负载。

它节省了带宽，jQuery框架从这些CDN加载得更快。

如果用户定期访问使用任何这些CDN的jQuery框架的站点，它将被缓存。

1. jQuery中有哪几种类型的选择器？（必须）

基本选择器

id选择器

class选择器

element选择器

\* 通配符选择器

selector1,selector1多条件选择器

层级选择器

父 后代选择器

父 直接子元素选择器

目标元素 + 紧相邻同辈下一个选择器

目标元素 ~ 紧相邻同辈所有选择器

属性选择器

[属性名]选择器

[属性名="属性值"]选择器

[属性名^="属性值"] 以某属性值开头 选择器

[属性名$="属性值"] 以某属性值结尾 选择器

[属性名\*="属性值"] 包含某属性值 选择器

1. jQuery中$符命名冲突解决方案?（必须）

释放$符的使用权，使用jQuery代替$

使用jQuery.noConflict()自定义一个符号,如：

var #$ = jQuery.noConflict()

1. jQuery中的选择器 和css中的选择器有区别吗？

jQuery选择器支持CSS里的选择器，

jQuery选择器可用来添加样式和添加相应的行为

CSS中的选择器是只能添加相应的样式

1. jQuery中如何来获取或和设置属性？

jQuery中可以用attr()方法来获取和设置元素属性

获取属性：$(this).attr('name')

设置属性: $(this).attr('name','val')

使用removeAttr() 方法来删除元素属性

1. jQuery中的hover()和toggle()有什么区别？

hover(fn1,fn2)：当鼠标移动到一个匹配的元素上面时，会触发指定的第一个函数。当鼠标移出这个元素时，会触发指定的第二个函数。

toggle(evenFn,oddFn): 每次点击时切换要调用的函数。如果点击了一个匹配的元素，则触发指定的第一个函数，当再次点击同一元素时，则触发指定的第二个函数。随后的每次点击都重复对这两个函数的轮番调用。

1. 介绍一下你所了解的遍历节点的常用方法?（必须）

children() 获取子元素,只考虑子元素不考虑后代元素

next() 获取下一个紧邻的兄弟元素

prev() 获取上一个紧邻的兄弟元素

siblings() 获取当前元素的所有兄弟元素（除了自己）

parents() 获取当前元素的所有祖先元素。

find() 取得匹配元素中的元素集合 包括子代和后代

1. jQuery 里的 each() 是什么函数？你是如何使用它的？

each() 函数就像是 Java 里的一个 Iterator，它允许你遍历一个元素集合。你可以传一个函数给 each() 方法，被调用的 jQuery 对象会在其每个元素上执行传入的函数。

1. jQuery.get() 和 jQuery.ajax() 方法之间的区别是什么?（必须）

ajax() 方法更强大，更具可配置性, 让你可以指定等待多久，以及如何处理错误。get() 方法是一个只获取一些数据的专门化方法。

1. jQuery中 detach() 和 remove() 方法的区别是什么?（重要）

remove()与detach()方法都是从dom中删除所有的元素

两者的共同之处在于都不会把匹配的元素从jQuery对象中删除。

不同之处在于用remove()删除的元素，除了元素被保留，其他的在这个元素上的绑定事件等都会被移除，但是detach()方法不会移除这个元素上的方法或者是附加数据。

1. 在一个 jQuery事件处理程序里返回了 false 会怎样？（重要）

这通常用于阻止事件向上冒泡。

1. 简述下为何通过ajax发送的请求会出现乱码问题，如何解决？（重要）

乱码的问题就是编码格式冲突，我们需要传输中文数据前面加一个encodeURIComponent()编码，例如：encodeURIComponent($("#name").val())；在接受参数的页面对传过来的编码过后的内容用后端语言进行解码

1. 如何找到所有 HTML select 标签的选中项？

使用:selected选择器，代码如下

$('[name="hobby"]:selected').val();

1. jQuery 中的方法链（链式操作）是什么？使用方法链有什么好处？

方法链是对一个方法返回的结果调用另一个方法，这使得代码简洁明了，同时由于只对 DOM 进行了一轮查找，性能方面更加出色。

1. 哪种方式更高效：document.getElementbyId("myId") 还是 $("#myId")？（必须）

第一种，因为它直接调用了 JavaScript 引擎。

1. 你是如何将一个 HTML 元素添加到 DOM 树中的？

你可以用 jQuery 方法 appendTo() 将一个 HTML 元素添加到 DOM 树中。这是 jQuery 提供的众多操控 DOM 的方法中的一个。你可以通过 appendTo() 方法在指定的 DOM 元素末尾添加一个现存的元素或者一个新的 HTML 元素。

1. jQuery的源码看过吗？能不能简单说一下它的实现原理？（重要）

jQuery给我们带来了一个简洁方便的编码模型

创建jQuery对象;

直接使用jQuery对象的属性/方法/事件),

一个强悍的dom元素查找器($)

插件式编程接口(jQuery.fn)

以及插件初始化的”配置”对象思想

1. jQuery操作样式css方法？

$(this).css('name') 如果括号内只有一个参数时则表示获取样式的值。

$(this).css('name','val') 如果有两个参数表示设置样式` 前面是属性名，后面是属性值 。

通过添加类,切换类,删除类的方式来操作css样式

css类操作

添加类名

$('this').addClass('className');

删除类名

$('this').removeClass('className');

切换类名

$('this').toggleClass('className');

//自动切换类名，有就删除 没有就添加

1. jquery的原理？（重要）

JQ本身就是一个类，在外面使用的$和jQuery是同一个东西，JQ中提供的方法分为两部分：

1. 写在jQuery原型上的方法，专门给JQ的实例使用
2. 写在jQuery私有属性上的，通过$.xxx可以获取进行操作等 jQuery() / $() =>创建JQ的实例，  
    需要传递两个参数，第一个参数一般是选择器内容；第二个参数是获取的上下文，如果不传递默  
    认是document；==>“此操作通俗的叫法：通过JQ选择器获取元素”
3. 返回结果是一个类数组(它也是JQ的实例)，这个类数组是JQ自己去创建的，里面有一些自己特  
    定的属性：length/context/selector/prevObject…
4. 获取到的结果我们叫做JQ对象(JQ实例)，可以调取JQ原型上提供的方法，但是它不是原生的JS  
    对象，不能调取浏览器提供的默认属性方法，当然原生JS也不能调用JQ上提供的属性方法；

Vue专题

1. 谈谈你对vue的理解? （重要）

Vue是一款轻量级框架，相比较于React，Angular等重量级框架而言，vue上手较为容易。

Vue是一套构建用户界面的渐进式框架,也可以理解为是一个视图模板引擎,强调的是状态到界面的映射。双向数据绑定是其最大的特点。

1. 什么是渐进式？

简单来说，渐进式可以近似理解为自由,你可以使用我的一部分，而不是说只要使用了一部分就要全部使用，可以根据自己的选择决定以何种形式使用，这也是vue框架的一大特点。

渐进式:声明式渲染 -> 组件系统 -> 客户端路由 ->集中式状态管理(vuex)->项目构建。

1. vue的特点？（重要）

易用：熟悉html，css，js可快速上手，

灵活性：在一个库和一套完整框架之间自由收缩

高效：20kb运行大小，超快虚拟Dom。

双向数据绑定：视图层（UI）与模型层(数据层，Model)双向绑定数据，当视图层发生改变时会动态的改变模型层的数据，当模型层发生改变时视图层也会响应的改变。

组件化开发:vue中对于一个完整的页面可以拆分成各个组件，就像是乐高积木一样，多个组件组成一个模型。

虚拟Dom:简单来说虚拟Dom就是通过js预先进行计算形成一个虚拟的Dom结构，并且该结构是一个优化后的结构，但是并没有真正的操作Dom，它只是虚拟的存在，等到所有计算完成之后再真正的Dom树上。

1. 什么是MVVM？（重要）

MVVM是Model-View-ViewModel的缩写,MVVM将视图与数据相分离，ViewModel则是二者之间的通信桥梁，它实时监控着视图层与数据层的变化，当视图层发生了修改时，ViewModel会将修改的信息动态的传递到数据层，数据层也会发生相应的修改；当数据层的数据发生变动时，ViewModel也会其相应到视图层，视图层对应的UI组件也会进行更新渲染。

1. vue双向绑定原理? （必须）

vue数据双向绑定是通过数据劫持结合发布者-订阅者模式的方式来实现。

数据劫持通过Object.defineProperty()来实现。

发布者-订阅模式双向数据绑定。

步骤如下：

1. 实现一个监听器Observer，用来劫持并监听所有属性，如果有变动的，就通知订阅者。
2. 实现一个订阅者Watcher，可以收到属性的变化通知并执行相应的函数，从而更新视图。
3. 实现一个解析器Compile，可以扫描和解析每个节点的相关指令，并根据初始化模板数据以及初  
    始化相应的订阅器。
4. vue双向绑定原理很重要，详解可参考



1. Vue中引入组件的步骤? （重要）

注册组件

Vue.component('my-component',{   
template: '<div>A custom component!</div>'  
})

导入组件:采用ES6的import ... from ...语法或者CommonJS的require()方法引入组件

使用组件：在需要的位置插入组件即可,如：

1. 局部组件与全局组件之间的区别? （重要）

局部组件必须在定义该组件的父组件中使用，比如：我在app这个根组件中定义的子组件，那就只能在app组件内使用，在其他地方无法使用。

1. 组件之间的通信方式？（必须）
2. 基于vuex实现通信：

使用Vuex存储数据实现数据共享,但是一般vuex存放的数据都是数据量较大，并且多个组件需要共享的数据，组件之间使用的数据定义在data中即可。

1. 普通组件之间实现通信

父组件向子组件传递数据首先在子组件内部添加一个props选项，里面存放一个数组用于接受父组件传递进来的值。

子组件中通过自定义事件向父组件传递信息 通过this.$emit('函数','参数')实现通信

非父子组件之间通过事件中心EventHub,hub.$on监听事件，hub.$emit()触发事件实现通信。

1. 什么是Vue生命周期?（必须）

简单来说vue生命周期就是一个vue实例从创建到销毁的一个过程，大致可分为实例初始化前,实例初始化后,挂载前,挂载后，更新前，更新后,销毁前,销毁后八个阶段，每个阶段对应一个相应生命周期钩子函数，不同阶段我们可以通过钩子函数来进行不同的操作。如果我们需要一个实例在销毁后再次出现的话，我们可以使用activated和deactivated配合keep-alive标签实现。

1. 列举出几个Vue中常用的生命周期钩子函数? （必须）

created: 实例已经创建完成之后调用,在这一步,实例已经完成数据观测, 属性和方法的运算, watch/event事件回调. 然而, 挂载阶段还没有开始, $el属性目前还不可见

mounted: el被新创建的 vm.$el 替换，并挂载到实例上去之后调用该钩子。如果 root实例挂载了一个文档内元素，当 mounted 被调用时 vm.$el 也在文档内。

activated::keep-alive组件激活时调用

1. 说出至少几种vue当中的指令和它的用法？（必须）

v-if：判断是否隐藏；

v-for：数据循环出来；

v-bind:class：绑定一个属性；

v-model：实现双向绑定

v-once： 只编译一次，显示内容之后不在具有响应式的功能

1. v-if与v-show的区别？（必须）

v-if是通过控制dom节点的存在与否来控制元素的显隐；

v-show是通过设置DOM元素的display样式，block为显示，none为隐藏

v-if判断是否加载，可以减轻服务器的压力，在需要时加载,但有更高的切换开销;v-show调整DOM元素的CSS的dispaly属性，可以使客户端操作更加流畅，但有更高的初始渲染开销。如果需要非常频繁地切换，则使用 v-show 较好；如果在运行时条件很少改变，则使用 v-if 较好。

1. vue中key值的作用？（必须）

key 的特殊属性主要用在 Vue 的虚拟 DOM 算法，在新旧 nodes 对比时辨识 VNodes。如果不使用 key，Vue 会使用一 种最大限度减少动态元素并且尽可能的尝试修复/再利用相 同类型元素的算法。使用 key，它会基于 key 的变化重新排 列元素顺序，并且会移除 key 不存在的元素。

1. vue中子组件如何调用父组件中的方法? （必须）

子组件调用父组件的方法可以使用 this.$emit()

1. <keep-alive></keep-alive>标签的作用是什么? （必须）

<keep-alive></keep-alive>包裹动态组件时，会缓存不活 动的组件实例,主要用于保留组件状态或避免重新渲染。比如有一个列表和一个详情，那么用户就会经常执行打开详情=>返回列表=>打开详情…这样的话列表和详情都是一个频率很高的页面，那么就可以对列表组件使用<keep-alive></keep-alive>进行缓存，这样用户每次返回列表的时候，都能从缓存中快速渲染，而不是重新渲染

1. vuex里面怎么分类别管理多项数据?

通过vuex里面的moudle功能，可以实现多模块划分。

1. 简述vuex?（必须）

Vuex 可以被看作项目中所有组件的数据中心,我们将所有组 件中共享的State抽离出来,任何组件都可以访问和操作我们 的数据中心。

1. Vuex的组成？（重要）

Vuex的组成：一个实例化的 Vuex.Store 由 state, mutations 和 actions三个属性组成。

state 中保存着共有数据

改变 state 中的数据有且只有通过mutations中的方法,且 mutations 中的方法必须是同步的

如果要写异步的方法,需要些在 actions中, 并通过 commi t 到 mutations 中进行 state 中数据的更改

1. jquery和vue的区别？（重要）

Jquery是单纯的js库，它是对原生js的一种封装集合，它的主要思想还是对dom 进行操作，但是vue则不同，vue作为新兴的js库但是它功能却不仅仅是一个库 所能概括，首先从根本的思想上来说，vue将不在进行繁琐的dom操作，基于 MVVM思想，vue专注于模型的ViewModel层，它将视图层（html-css）与Model(js) 完全剥离出来，通过VM作为二者之间的通信桥梁，当Model层发生改变时，view 层也会进行响应的跟新渲染，当view层修改了数据时，model层中的数据也会 发生改变，也就是我们所说的数据双向绑定。

1. vue有那些技术栈？

vuejs 基础库

vue-router 路由

vue-vuex 存储数据流

vue-cli 命令行

vue-devtools 开发调试工具

vue-press 静态文档写作

1. 使用插值绑定数据时，刷新页面时会闪现{{}}，如何解决？（重要）

v-text方案

<h2 v-text="message"></h2>

v-cloak方案

<style>

[v-cloak]{

display: none;

}

</style>

<h2 v-cloak>{{message}}</h2>

1. v-text和v-html的区别？什么xss攻击？（重要）

v-text不能渲染html元素，他会以字符实体的形式渲染出来，v-html可以渲染 html元素，但是这种方式存在安全隐患(跨域脚本攻击)。Xss是一种跨域脚本攻 击，它的本质就是利用web应用的漏洞向页面中嵌入一段脚本(特殊字符如<>浏 览器会解析为html标签)，此时浏览器会认为是用户动态加入进而执行导致破坏 网站，修改路由等。

1. vue-loader是什么？使用它的用途有哪些？

解析.vue文件的一个加载器，跟template/js/style转换成js模块。

用途：js可以写es6、style样式可以scss或less、template可以加jade等

1. 请介绍一下key属性 （重要）

Vue会复用已渲染的元素而不是从头开始渲染，从而达到高效渲染的目的，这种 情况有一定优势就是在于两个相同的模板当二者不需要相对独立时，就可以通过 改变两个模板中的不同信息来实现模板的切换，但是实际上只渲染了一个模板。 对于相对独立的两个元素来说就不能让他们被复用，这时就需要使用key帮助 vue进行区分，key是唯一的，类似于id不可重复。

1. v-model是什么？怎么使用？ vue中标签怎么绑定事件？（重要）

可以实现双向绑定，指令（v-class、v-for、v-if、v-show、v-on）。vue的model层的data属性。绑定事件：<input @click=doLog() />

1. axios是什么？怎么使用？描述使用它实现登录功能的流程？（重要）

请求后台资源的模块。npm install axios -S装好，然后发送的是跨域，需在配置文件中config/index.js进行设置。后台如果是Tp5则定义一个资源路由。js中使用import进来，然后.get或.post。返回在.then函数中如果成功，失败则是在.catch函数中

1. 什么是RESTful API？怎么使用? （重要）

是一个api的标准，无状态请求。请求的路由地址是固定的，如果是tp5则先路由配置中把资源路由配置好。标准有：.post .put .delete

1. 导航钩子有哪些？它们有哪些参数？ （重要）

导航钩子有：a/全局钩子和组件内独享的钩子。b/beforeRouteEnter、afterEnter、beforeRouterUpdate、beforeRouteLeave

参数：有to（去的那个路由）、from（离开的路由）、next（一定要用这个函数才能去到下一个路由，如果不用就拦截）最常用就这几种

1. 请详细说下你对vue生命周期的理解？（必须）

总共分为8个阶段创建前/后，载入前/后，更新前/后，销毁前/后。

创建前/后：

在beforeCreated阶段，vue实例的挂载元素$el和数据对象data都为undefined，还未初始化。在created阶段，vue实例的数据对象data有了，$el还没有。

载入前/后：

在beforeMount阶段，vue实例的$el和data都初始化了，但还是挂载之前为虚拟的dom节点，data.message还未替换。在mounted阶段，vue实例挂载完成，data.message成功渲染。

更新前/后：

当data变化时，会触发beforeUpdate和updated方法。

销毁前/后：

在执行destroy方法后，对data的改变不会再触发周期函数，说明此时vue实例已经解除了事件监听以及和dom的绑定，但是dom结构依然存在

1. 说出vue.cli项目中src目录每个文件夹和文件的用法？

assets文件夹是放静态资源；components是放组件；router是定义路由相关的配置;view视图；app.vue是一个应用主组件；main.js是入口文件

1. vue-router的使用及实现原理？（重要）

前端路由是直接找到与地址匹配的一个组件或对象并将其渲染出来。改变浏览器地址而不向服务器发出请求有两种方式:

1. 在地址中加入#以欺骗浏览器，地址的改变是由于正在进行页内导航

2. 使用H5的window.history功能，使用URL的Hash来模拟一个完整的URL。

当打包构建应用时，Javascript 包会变得非常大，影响页面加载。如果我们能把不同路由对应的组件分割成不同的代码块，然后当路由被访问的时候才加载对应组件，这样就更加高效了。

1. vuex有哪几种属性？（重要）

有五种，分别是 State、 Getter、Mutation 、Action、 Module

1. vuex的State特性是？
2. Vuex就是一个仓库，仓库里面放了很多对象。其中state就是数据源存放地，对应于与一般Vue  
    对象里面的data
3. state里面存放的数据是响应式的，Vue组件从store中读取数据，若是store中的数据发生改变，  
    依赖这个数据的组件也会发生更新
4. 它通过mapState把全局的 state 和 getters 映射到当前组件的 computed 计算属性中
5. vuex的Getter特性是？
6. getters 可以对State进行计算操作，它就是Store的计算属性
7. 虽然在组件内也可以做计算属性，但是getters 可以在多组件之间复用
8. 如果一个状态只在一个组件内使用，是可以不用getters
9. vuex的Mutation特性是？
10. Action 类似于 mutation，不同在于：
11. Action 提交的是 mutation，而不是直接变更状态。
12. Action 可以包含任意异步操作
13. Vue.js中ajax请求代码应该写在组件的methods中还是vuex的actions中？
14. 如果请求来的数据是不是要被其他组件公用，仅仅在请求的组件内使用，就不需要放入vuex 的  
     state里。
15. 如果被其他地方复用，这个很大几率上是需要的，如果需要，请将请求放入action里，方便复用，并包装成promise返回，在调用处用async await处理返回的数据。如果不要复用这个请求，那么直接写在vue文件里很方便。
16. 不用Vuex会带来什么问题？（必须）
17. 可维护性会下降，你要想修改数据，你得维护三个地方
18. 可读性会下降，因为一个组件里的数据，你根本就看不出来是从哪来的
19. 增加耦合，大量的上传派发，会让耦合性大大的增加，本来Vue用Component就是为了减少耦  
     合，现在这么用，和组件化的初衷相背。
20. vue生命周期的作用是什么？（必须）

它的生命周期中有多个事件钩子，让我们在控制整个Vue实例的过程时更容易形成好的逻辑。

1. 第一次页面加载会触发哪几个钩子？（必须）

第一次页面加载时会触发 beforeCreate, created, beforeMount, mounted 这几个钩子

1. DOM 渲染在 哪个周期中就已经完成？（必须）

DOM 渲染在 mounted 中就已经完成了。

1. 简单描述每个周期具体适合哪些场景？（必须）

生命周期钩子的一些使用方法：

beforecreate : 可以在这加个loading事件，在加载实例时触发

created : 初始化完成时的事件写在这里，如在这结束loading事件，异步请求也适宜在这里调用

mounted : 挂载元素，获取到DOM节点

updated : 如果对数据统一处理，在这里写上相应函数

beforeDestroy : 可以做一个确认停止事件的确认框

nextTick : 更新数据后立即操作dom

1. axios的特点有哪些？（必须）
2. Axios 是一个基于 promise 的 HTTP 库，支持promise所有的API
3. 它可以拦截请求和响应
4. 它可以转换请求数据和响应数据，并对响应回来的内容自动转换成 JSON类型的数据
5. 安全性更高，客户端支持防御 XSRF
6. axios有哪些常用方法？（必须）
7. axios.get(url[, config]) //get请求用于列表和信息查询
8. axios.delete(url[, config]) //删除
9. axios.post(url[, data[, config]]) //post请求用于信息的添加
10. axios.put(url[, data[, config]]) //更新操作
11. v-show和v-if指令的共同点和不同点?（必须）

v-show指令是通过修改元素的displayCSS属性让其显示或者隐藏

v-if指令是直接销毁和重建DOM达到让元素显示和隐藏的效果

1. 如何让CSS只在当前组件中起作用?（重要）

将当前组件的<style>修改为<style scoped>

1. vue 对IE8兼容怎么样？（重要）

不兼容IE8

1. 请说下封装 vue 组件的过程？

首先，组件可以提升整个项目的开发效率。能够把页面抽象成多个相对独立的模块，解决了我们传统项目开发：效率低、难维护、复用性等问题。

然后，使用Vue.extend方法创建一个组件，然后使用Vue.component方法注册组件。子组件需要数据，可以在props中接受定义。而子组件修改好数据后，想把数据传递给父组件。可以采用emit方法。

1. 聊聊你对Vue.js的template编译的理解？

简而言之，就是先转化成AST树，再得到的render函数返回VNode（Vue的虚拟DOM节点）

详情步骤

首先，通过compile编译器把template编译成AST语法树（abstract syntax tree 即 源代码的抽象语法结构的树状表现形式），compile是createCompiler的返回值，createCompiler是用以创建编译器的。另外compile还负责合并option。

然后，AST会经过generate（将AST语法树转化成render funtion字符串的过程）得到render函数，render的返回值是VNode，VNode是Vue的虚拟DOM节点，里面有（标签名、子节点、文本等等）

1. 计算属性和 watch 的区别？（重要）

计算属性是自动监听依赖值的变化，从而动态返回内容，监听是一个过程，在监听的值变化时，可以触发一个回调，并做一些事情。

所以区别来源于用法，只是需要动态值，那就用计算属性；需要知道值的改变后执行业务逻辑，才用 watch，用反或混用虽然可行，但都是不正确的用法。

总结

当有一些数据需要随着另外一些数据变化时，建议使用computed。

当有一个通用的响应数据变化的时候，要执行一些业务逻辑或异步操作的时候建议使用watcher

1. $nextTick的使用？（必须）

当你修改了data的值然后马上获取这个dom元素的值，是不能获取到更新后的值，

你需要使用$nextTick这个回调，让修改后的data值渲染更新到dom元素之后在获取，才能成功。

1. 理解Vue中的Render渲染函数？（必须）

VUE一般使用template来创建HTML，然后在有的时候，我们需要使用javascript来创建html，这时候我们需要使用render函数。

1. vue组件中data为什么必须是一个函数？（重要）

因为JavaScript的特性所导致，在component中，data必须以函数的形式存在，不可以是对象。

组建中的data写成一个函数，数据以函数返回值的形式定义，这样每次复用组件的时候，都会返回一份新的data，相当于每个组件实例都有自己私有的数据空间，它们只负责各自维护的数据，不会造成混乱。而单纯的写成对象形式，就是所有的组件实例共用了一个data，这样改一个全都改了。

1. 渐进式框架的理解？

主张最少；可以根据不同的需求选择不同的层级；

1. 单页面应用和多页面应用区别及优缺点？（重要）

单页面应用（SPA），通俗一点说就是指只有一个主页面的应用，浏览器一开始要加载所有必须的 html, js, css。所有的页面内容都包含在这个所谓的主页面中。但在写的时候，还是会分开写（页面片段），然后在交互的时候由路由程序动态载入，单页面的页面跳转，仅刷新局部资源。多应用于pc端。

多页面（MPA），就是指一个应用中有多个页面，页面跳转时是整页刷新

单页面的优点

用户体验好，快，内容的改变不需要重新加载整个页面，基于这一点spa对服务器压力较小；前后端分离；页面效果会比较炫酷（比如切换页面内容时的专场动画）。

单页面缺点

不利于seo；导航不可用，如果一定要导航需要自行实现前进、后退。（由于是单页面不能用浏览器的前进后退功能，所以需要自己建立堆栈管理）；初次加载时耗时多；页面复杂度提高很多。

1. v-if和v-for的优先级？（重要）

当 v-if 与 v-for 一起使用时，v-for 具有比 v-if 更高的优先级，这意味着 v-if 将分别重复运行于每个 v-for 循环中。所以，不推荐v-if和v-for同时使用。

如果v-if和v-for一起用的话，vue中的的会自动提示v-if应该放到外层去。

1. vue常用的修饰符？（重要）

.stop：等同于JavaScript中的event.stopPropagation()，防止事件冒泡；

.prevent：等同于JavaScript中的event.preventDefault()，防止执行预设的行为（如果事件可取消，  
则取消该事件，而不停止事件的进一步传播）；

.capture：与事件冒泡的方向相反，事件捕获由外到内；

.self：只会触发自己范围内的事件，不包含子元素；

.once：只会触发一次。

1. vue的两个核心点?（必须）

数据驱动、组件系统

数据驱动：ViewModel，保证数据和视图的一致性。

组件系统：应用类UI可以看作全部是由组件树构成的。

1. SPA首屏加载慢如何解决? （重要）

使用懒加载（按需加载）技术；使用CDN资源。

1. Vue-router跳转和location.href有什么区别?

使用location.href='/url'来跳转，简单方便，但是刷新了页面；

使用history.pushState('/url')，无刷新页面，静态跳转；

引进router，然后使用router.push('/url')来跳转，使用了diff算法，实现了按需加载，减少了dom的消耗。

其实使用router跳转和使用history.pushState()没什么差别的，因为vue-router就是用了history.pushState()，尤其是在history模式下。

1. vue slot 插槽是什么？

简单来说，假如父组件需要在子组件内放一些DOM，那么这些DOM是显示、不显示、在哪个地方显示、如何显示，就是slot分发负责的活。

1. params和query的区别？

用法：query要用path来引入，params要用name来引入，接收参数都是类似的，分别是this.$route.query.name和this.$route.params.name。

url地址显示：query更加类似于我们ajax中get传参，params则类似于post，说的再简单一点，前者在浏览器地址栏中显示参数，后者则不显示

注意点：query刷新不会丢失query里面的数据

params刷新 会 丢失 params里面的数据。

1. vue更新数组时触发视图更新的方法？（必须）

push()；

pop()；

shift()；

unshift()；

splice()；

sort()；

reverse()

1. created和mounted的区别？（必须）

created:在模板渲染成html前调用，即通常初始化某些属性值，然后再渲染成视图。

mounted:在模板渲染成html后调用，通常是初始化页面完成后，再对html的dom节点进行一些需要的操作。

1. mvvm 框架是什么？（必须）

vue是实现了双向数据绑定的mvvm框架，当视图改变更新模型层，当模型层改变更新视图层。在vue中，使用了双向绑定技术，就是View的变化能实时让Model发生变化，而Model的变化也能实时更新到View。

1. mvvm和mvc区别？

mvc和mvvm其实区别并不大，都是一种设计思想。主要就是mvc中Controller演变成mvvm中的viewModel。mvvm主要解决了mvc中大量的DOM 操作使页面渲染性能降低，加载速度变慢，影响用户体验。

区别：vue数据驱动，通过数据来显示视图层而不是节点操作。

场景：数据操作比较多、频繁的场景，更加便捷。

1. $route 和 $router 的区别？（重要）

$router是VueRouter的实例，在script标签中想要导航到不同的URL,使用$router.push方法。返回上一个历史history用$router.to(-1)

$route为当前router跳转对象。里面可以获取当前路由的name,path,query,parmas等。

1. 路由之间跳转？

声明式（标签跳转 router-link）

编程式（ js跳转 this.$router.push）

1. vue-router的两种模式？（重要）

hash模式：即地址栏 URL 中的 # 符号；

history模式：window.history对象打印出来可以看到里边提供的方法和记录长度。利用了 HTML5 History Interface 中新增的 pushState() 和 replaceState() 方法。（需要特定浏览器支持）。

1. vue如何获取dom? （重要）

ref="domName" 用法：this.$refs.domName

1. v-on可以监听多个方法吗？

可以，具体如下：

<input type="text" v-on="{ input:onInput,focus:onFocus,blur:onBlur, }">。

1. vue初始化页面闪动问题？

使用vue开发时，在vue初始化之前，由于div是不归vue管的，所以我们写的代码在还没有解析的情况下会容易出现花屏现象，看到类似于{{message}}的字样，虽然一般情况下这个时间很短暂，但是我们还是有必要让解决这个问题的。

首先：在css里加上[v-cloak] {

display: none;

}。

1. active-class是哪个组件的属性？

vue-router模块的router-link组件。

1. 说出至少4种vue当中的指令和它的用法？

v-if：判断是否隐藏；

v-for：数据循环；

v-bind:class：绑定一个属性；

v-model：实现双向绑定；

1. 怎么定义vue-router的动态路由？怎么获取传过来的动态参数？

在router目录下的index.js文件中，对path属性加上/:id。

使用router对象的params.id。

1. vue-router有哪几种导航钩子？（重要）

第一种：是全局导航钩子：router.beforeEach(to,from,next)，作用：跳转前进行判断拦截。

第二种：组件内的钩子

第三种：单独路由独享组件

1. 说一下什么是Virtual DOM?（重要）

Virtual DOM 是 DOM 节点在 JavaScript 中的一种抽象数据结构，之所以需要虚拟DOM，是因为浏览器中操作DOM的代价比较昂贵，频繁操作DOM会产生性能问题。虚拟DOM的作用是在每一次响应式数据发生变化引起页面重渲染时，Vue对比更新前后的虚拟DOM，匹配找出尽可能少的需要更新的真实DOM，从而达到提升性能的目的。

1. 介绍一下Vue中的Diff算法? （必须）

在新老虚拟DOM对比时

1. 对比节点本身，判断是否为同一节点，如果不为相同节点，则删除该节点重新创建节点进行替换
2. 如果为相同节点，进行patchVnode，判断如何对该节点的子节点进行处理，先判断一方有子节点一方没有子节点的情况(如果新的children没有子节点，将旧的子节点移除)

比较如果都有子节点，则进行updateChildren，判断如何对这些新老节点的子节点进行操作（diff核心）。

1. 匹配时，找到相同的子节点，递归比较子节点

在diff中，只对同层的子节点进行比较，放弃跨级的节点比较，使得时间复杂从O(n^3)降低值O(n)，也就是说，只有当新旧children都为多个子节点时才需要用核心的Diff算法进行同层级比较。

1. 说说Vue2.0和Vue3.0有什么区别? （必须）
2. 重构响应式系统，使用Proxy替换Object.defineProperty，使用Proxy优势：
3. 可直接监听数组类型的数据变化
4. 监听目标为对象本身，不需要像Object.defineProperty一样遍历每个属性，有一定的性能提升
5. 可拦截apply、ownKeys、has等13种方法，而Object.defineProperty不行
6. 直接实现对象属性的新增/删除
7. 新增Composition API，更好的逻辑复用和代码组织
8. 重构 Virtual DOM
9. 模板编译时的优化，将一些静态节点编译成常量
10. slot优化，将slot编译为lazy函数，将slot的渲染的决定权交给子组件
11. 模板中内联事件的提取并重用（原本每次渲染都重新生成内联函数）
12. 代码结构调整，更便于Tree shaking，使得体积更小
13. 使用Typescript替换Flow
14. 为什么要新增Composition API，它能解决什么问题？（重要）

Vue2.0中，随着功能的增加，组件变得越来越复杂，越来越难维护，而难以维护的根本原因是Vue的API设计迫使开发者使用watch，computed，methods选项组织代码，而不是实际的业务逻辑。

另外Vue2.0缺少一种较为简洁的低成本的机制来完成逻辑复用，虽然可以minxis完成逻辑复用，但是当mixin变多的时候，会使得难以找到对应的data、computed或者method来源于哪个mixin，使得类型推断难以进行。

所以Composition API的出现，主要是也是为了解决Option API带来的问题，第一个是代码组织问题，Compostion API可以让开发者根据业务逻辑组织自己的代码，让代码具备更好的可读性和可扩展性，也就是说当下一个开发者接触这一段不是他自己写的代码时，他可以更好的利用代码的组织反推出实际的业务逻辑，或者根据业务逻辑更好的理解代码。

第二个是实现代码的逻辑提取与复用，当然mixin也可以实现逻辑提取与复用，但是像前面所说的，多个mixin作用在同一个组件时，很难看出property是来源于哪个mixin，来源不清楚，另外，多个mixin的property存在变量命名冲突的风险。而Composition API刚好解决了这两个问题。

1. 都说Composition API与React Hook很像，说说区别？（重要）

从React Hook的实现角度看，React Hook是根据useState调用的顺序来确定下一次重渲染时的state是来源于哪个useState，所以出现了以下限制

不能在循环、条件、嵌套函数中调用Hook

必须确保总是在你的React函数的顶层调用Hook

useEffect、useMemo等函数必须手动确定依赖关系

而Composition API是基于Vue的响应式系统实现的，与React Hook的相比

声明在setup函数内，一次组件实例化只调用一次setup，而React Hook每次重渲染都需要调用Hook，使得React的GC比Vue更有压力，性能也相对于Vue来说也较慢

Compositon API的调用不需要顾虑调用顺序，也可以在循环、条件、嵌套函数中使用

响应式系统自动实现了依赖收集，进而组件的部分的性能优化由Vue内部自己完成，而React Hook需要手动传入依赖，而且必须必须保证依赖的顺序，让useEffect、useMemo等函数正确的捕获依赖变量，否则会由于依赖不正确使得组件性能下降。

虽然Compositon API看起来比React Hook好用，但是其设计思想也是借鉴React Hook的。

1. 区分Real 真实DOM和Virtual虚拟 DOM？（ 重点 ）

Real DOM

1. 更新缓慢。
2. 可以直接更新 HTML。
3. 如果元素更新，则创建新DOM。
4. DOM操作代价很高。
5. 消耗的内存较多。

Virtual DOM

1. 更新更快。
2. 无法直接更新 HTML
3. 如果元素更新，则更新 JSX
4. DOM 操作非常简单
5. 很少的内存消耗
6. 那template和jsx的有什么分别？ （重要）

对于 runtime 来说，只需要保证组件存在 render 函数即可，而我们有了预编译之后，我们只需要保证构建过程中生成 render 函数就可以。

在 webpack 中，我们使用vue-loader编译.vue文件，内部依赖的vue-template-compiler模块，在 webpack 构建过程中，将template预编译成 render 函数。

与 react 类似，在添加了jsx的语法糖解析器babel-plugin-transform-vue-jsx之后，就可以直接手写render函数。

所以，template和jsx的都是render的一种表现形式，不同的是：

JSX相对于template而言，具有更高的灵活性，在复杂的组件中，更具有优势，而 template 虽然显得有些呆滞。但是 template 在代码结构上更符合视图与逻辑分离的习惯，更简单、更直观、更好维护。

项目专题

1. 请简述一下数据库的优化?

数据库的优化可以从四个方面来优化：

1. 从结构层: web服务器采用负载均衡服务器,mysql服务器采用主从复制,读写分离
2. 从储存层: 采用合适的存储引擎,采用三范式
3. 从设计层: 采用分区分表,索引,表的字段采用合适的字段属性,适当的采用逆范式,开启mysql缓存
4. sql语句层:结果一样的情况下,采用效率高,速度快节省资源的sql语句执行
5. 购物车的原理？

购物车相当于现实中超市的购物车，不同的是一个是实体车，一个是虚拟车而已。用户可以在购物网站的不同页面之间跳转，以选购自己喜爱的商品，点击购买时，该商品就自动保存到你的购物车中，重复选购后，最后将选中的所有商品放在购物车中统一到付款台结账，这也是尽量让客户体验到现实生活中购物的感觉。服务器通过追踪每个用户的行动，以保证在结账时每件商品都物有其主。

主要涉及以下几点:

1、把商品添加到购物车，即订购

2、删除购物车中已定购的商品

3、修改购物车中某一本图书的订购数量

4、清空购物车

5、显示购物车中商品清单及数量、价格

实现购物车的关键在于服务器识别每一个用户并维持与他们的联系。但是HTTP协议是一种“无状态(Stateless)”的协议，因而服务器不能记住是谁在购买商品，当把商品加入购物车时，服务器也不知道购物车里原先有些什么，使得用户在不同页面间跳转时购物车无法“随身携带”，这都给购物车的实现造成了一定的困难。

1. 是否了解公钥加密和私钥加密。如何确保表单提交里的密码字段不被泄露?

公钥用于对数据进行加密，私钥用于对数据进行解密。

很直观的理解：公钥就是公开的密钥，其公开了大家才能用它来加密数据。私钥是私有的密钥，谁有这个密钥才能够解密密文。

解决方案 1

form 在提交的过程中，对密码字段是不进行加密而是以明码的形式进行数据传输的。 如果要对数据进行加密，你可以自己写一个脚本对内容进行编码后传输，只是这个安全性也并不高。

解决方案 2

如果想对数据进行加密，你可以使用 HTTPS 安全传输协议，这个协议是由系统进行密码加密处理的，在数据传输中是绝对不会被拦截获取的，只是 HTTPS 的架设会相对麻烦点。一些大型网站的登录、银行的在线网关等都是走这条路。

1. 验证码是干嘛的，是为了解决什么安全问题?

所谓验证码，就是将一串随机产生的数字或符号，生成一幅图片， 图片里加上一些干扰象素（防止OCR），由用户肉眼识别其中的验证码信息，输入表单提交网站验证，验证成功后才能使用某项功能。

验证码一般是防止批量注册的，人眼看起来都费劲，何况是机器。

像百度贴吧未登录发贴要输入验证码大概是防止大规模匿名回帖的发生。

目前，不少网站为了防止用户利用机器人自动注册、登录、灌水，都采用了验证码技术。

1. 日常前端代码开发中，有哪些值得用ES6去改进的编程优化或者规范？

常用箭头函数来取代var self = this;的做法。

常用let取代var命令。

常用数组/对象的结构赋值来命名变量，结构更清晰，语义更明确，可读性更好。

在长字符串多变量组合场合，用模板字符串来取代字符串累加，能取得更好地效果和阅读体验。

用Class类取代传统的构造函数，来生成实例化对象。

在大型应用开发中，要保持module模块化开发思维，分清模块之间的关系，常用import、export方法。

1. 如果已经开发完一个web应用用于手机端访问，请问需要在哪方面优化页面？
2. 按需加载
3. 静态资源压缩
4. 巧用图标字体
5. 保证首屏相应速度
6. 动画能用css3实现的尽量用css3实现
7. 移动端事件优化（fastclick）
8. 有没有什么问题要问我？

1 .别问加班和啥时候调工资之类的

1. 社保、公积金交么（千万别说基数是多少）
2. 问公司做的项目主要是哪方面呢，我擅长的技术\*\*\*\*（具体技术）不知道公司需不需要
3. 注意：如果去的是国企，国企很强调奉献，尽量往这方面靠
4. 面试公司前最好搜索下公司是干什么的，能把他们产品说上来点就更好了（上市公司、国企尤其  
    得这样）
5. 我们这边的话会有加班的情况，不知道你能接受不？

可以，个人愿意与公司一块成长的，而且也有了解这行会有加班，会配合公司把项目用心完成的。

1. 为啥要离开上家单位呀（或者从前从事的不是计算机相关，为啥改行了）？
2. 千万别说这行工资高之类的
3. 要说，自己对这个很感兴趣，大学的时候加入的有这个社团，跟着学长学了这个东西，越学越感  
    兴趣，于是利用大学课余时间自学了前端的知识（也可以把这个东西说成自己亲戚）
4. 你都做过什么项目呢？让他具体聊某一个项目中运用的技术？

注意：用心找自己做的项目中自己感觉最拿出来手的（复杂度最高，用的技术最多的项目），描述的时候尽可能往里面添加一些技术名词

1．布局我们用html5+css3

2. 我们会用reset.css重置浏览器的默认样式

3. JS框架的话我们选用的是jQuery(也可能是Zepto)

4. 我们用版本控制工具git来协同开发

5. 我们会基于gulp搭建的前端自动化工程来开发（里面包含有我们的项目结构、我们需要引用的  
第三方库等一些信息，我们还实现了sass编译、CSS3加前缀等的自动化）

6. 我们的项目中还用到了表单验证validate插件、图片懒加载Lazyload插件

1. 专业宽度方面：最近在学习哪些新技术呀，以后准备往哪方面发展？

我最近在学习react …

我最近在学习sea.js …

我最近在学require.js …

我最近正在恶补node.js …

技术方向：高级前端开发工程师、全栈工程师（前端+后端nodejs）、前端架构师

管理方向：前端leader(前端主管、前端经理)、项目经理、CTO

1. 平常碰到问题一般都是怎么解决的呢？

百度、看一些博客（csdn、博客园cnblogs）、去论坛提问（比如知乎）、开发者问答社区（segmentfault）、会加一些qq技术交流群

看前端大牛的博客

玉伯 kissy框架（PC端淘宝、天猫）,sea.js（前端模块加载器）

司徒正美 avalon（前端MVC框架）

阮一峰（翻译了很多国外文档）

张鑫旭（写了大量的博客，用生活化的例子来解释难点）

大漠穷秋（angular1，angular2慕课网视频教程）

徐飞（技术栈）

朴灵

寸志

1. 你有跟后端合作的经验是吧，你们都怎么协作的呀？

第一种回答：（smarty啥都描述不出来，ajax交互这块特别特别差的学生，但这限定了你能就业的公司，因为大部分公司还是希望他招的人是很全面的人才，所以这个不推荐）

我们公司前端代码做完之后，直接就把前端代码的话交给后端了，页面绑定数据和ajax这块基本上都是后端开发工程师来完成的。除非他们碰到一些复杂点的js特别难写或者有些js的话需要处理样式这块的话会来让我们弄。

第二种回答：（适合大部分同学，前提你得理清楚）

我们公司的话，后端用的php语言，平时的话我们也会套一些smarty模板，只要我们前后端约定一下数据格式，然后我按照smarty模板的一些语法把数据解析成最终的HTML。

另外我们公司ajax这块的话一般都由我们前端来做，而且后端做的ajax这块我们通常也会去看下，尤其是ajax调用成功还需要js操作css样式这块的情况。Ajax接口这块的话，我们会事先约定下数据格式，我们公司的接口一般都是json格式的，而且我们有个文档是专门描述这个接口（里面通常会描述下返回结果是对象还是数组，对应里面的每一项又都是啥），我们前端只需要把json格式的数据结合上html渲染到我们也没对应的位置即可。我们拿到数据的话，现在一般很少直接用js拼接html了，一般会借助js模板引擎（handlerbars）来做这方面工作。

1. 简单描述一下你做过的项目研发流程？
2. 约定规范(目录结构、命名规范、代码书写规范)
3. 由我们组长（公司有的叫组长有的叫leader）给我们搭建前端工程gulp(最起码把browsersync  
    模块能说出来，用它创建一个局域网server并实现监控文件有变化就自动刷新功能)（另外搭建  
    前端工程的时候就会把我们会用到的库或者插件放进去）、拆分不同的前端模块给我们开发
4. 正式开发（我们通过git工具来进行）
5. 代码review（由我们组长来帮我们做）
6. 交付给后端开发工程师，需要帮他们联调的话到时再帮他们联调
7. 你们移动端怎么开发的？用的什么单位、js框架呢、怎么真机调试你们程序、怎么解决前缀？

首先我们移动端用的是rem这个单位，移动端实际上还新增了vw, vh等一些单位，但是相对于rem来说，他们的兼容性都不好。rem是一个相对单位，是相对于根节点的font-size的比例。我们还会引用一个外部的js，这个js可以通过屏幕宽度动态计算根节点的font-size值。

我们移动端用的JS框架是zepto.js，因为相应来说它的的体积的话要小很多。而且我们项目中有时会需要使用一些触屏事件，比如滑动事件，那么我们还会调用它里面的touch模块

我们webapp的话都是基于gulp搭建的前端工程来开发，因为利用gulp起一个webserver特别简单，并且我还可以实现livereload功能（当我监控的文件发生变化的时候，可以触发浏览器的自动刷新功能），而且在css3里面加前缀的话是一个很头疼的问题，我们现在只需要用autoprefixer模块就可以很容易实现自动加css3前缀这个功能了。

1. vuejs与angularjs以及react的区别？
2. 与AngularJS的区别

相同点：

都支持指令：内置指令和自定义指令。

都支持过滤器：内置过滤器和自定义过滤器。

都支持双向数据绑定。

都不支持低端浏览器。

不同点：

1. AngularJS的学习成本高，比如增加了Dependency Injection特性，而Vue.js本身提供的API都比较简单、直观。
2. 在性能上，AngularJS依赖对数据做脏检查，所以Watcher越多越慢。

Vue.js使用基于依赖追踪的观察并且使用异步队列更新。所有的数据都是独立触发的。

对于庞大的应用来说，这个优化差异还是比较明显的。

1. 与React的区别

相同点：

React采用特殊的JSX语法，Vue.js在组件开发中也推崇编写.vue特殊文件格式，对文件内容都有一些约定，两者都需要编译后使用。

中心思想相同：一切都是组件，组件实例之间可以嵌套。

都提供合理的钩子函数，可以让开发者定制化地去处理需求。

都不内置列数AJAX，Route等功能到核心包，而是以插件的方式加载。

在组件开发中都支持mixins的特性。

不同点：

React依赖Virtual DOM,而Vue.js使用的是DOM模板。React采用的Virtual DOM会对渲染出来的结果做脏检查。

Vue.js在模板中提供了指令，过滤器等，可以非常方便，快捷地操作Virtual DOM。

vue生命周期面试题

1. SSR有了解吗？原理是什么？

在客户端请求服务器的时候，服务器到数据库中获取到相关的数据，并且在服务器内部将Vue组件渲染成HTML，并且将数据、HTML一并返回给客户端，这个在服务器将数据和组件转化为HTML的过程，叫做服务端渲染SSR。

而当客户端拿到服务器渲染的HTML和数据之后，由于数据已经有了，客户端不需要再一次请求数据，而只需要将数据同步到组件或者Vuex内部即可。除了数据意外，HTML也结构已经有了，客户端在渲染组件的时候，也只需要将HTML的DOM节点映射到Virtual DOM即可，不需要重新创建DOM节点，这个将数据和HTML同步的过程，又叫做客户端激活。

使用SSR的好处：

1. 有利于SEO：其实就是有利于爬虫来爬你的页面，因为部分页面爬虫是不支持执行JavaScript的，这种不支持执行JavaScript的爬虫抓取到的非SSR的页面会是一个空的HTML页面，而有了SSR以后，这些爬虫就可以获取到完整的HTML结构的数据，进而收录到搜索引擎中。
2. 白屏时间更短：相对于客户端渲染，服务端渲染在浏览器请求URL之后已经得到了一个带有数据的HTML文本，浏览器只需要解析HTML，直接构建DOM树就可以。而客户端渲染，需要先得到一个空的HTML页面，这个时候页面已经进入白屏，之后还需要经过加载并执行 JavaScript、请求后端服务器获取数据、JavaScript 渲染页面几个过程才可以看到最后的页面。特别是在复杂应用中，由于需要加载 JavaScript 脚本，越是复杂的应用，需要加载的 JavaScript 脚本就越多、越大，这会导致应用的首屏加载时间非常长，进而降低了体验感。
3. 视差滚动效果？

视差滚动（Parallax Scrolling）通过在网页向下滚动的时候，控制背景的移动速度比前景的移动速度慢来创建出令人惊叹的3D效果。

CSS3实现

优点：开发时间短、性能和开发效率比较好，缺点是不能兼容到低版本的浏览器

jQuery实现

通过控制不同层滚动速度，计算每一层的时间，控制滚动效果。

优点：能兼容到各个版本的，效果可控性好

缺点：开发起来对制作者要求高

插件实现方式

例如：parallax-scrolling，兼容性十分好

1. 如果需要手动写动画，你认为最小时间间隔是多久，为什么？

多数显示器默认频率是60Hz，即1秒刷新60次，所以理论上最小间隔为1/60＊1000ms ＝ 16.7ms。

1. 222