

Version 2.1

Web 全栈面试宝典

无限互联

教学研发与运营中心

无限互联专有信息声明

本文档中所有信息均为无限互联机密，未经版权限定及星宇无限（北京）科技有限公司明确的书面许可，任何公司，组织和个人不得为任何目的、以任何手段或形式对本文档的任何部分或全部进行复制，存储，引入检索系统或传播。

虽然无限互联已经尽最大努力使本文档内容正确有效，但仍然可能有某些技术方面不够准确或存在印刷错误。这些信息将会被不定期的更新，更新的内容将包含在本文档的未来版本中。

“无限互联”、是星宇无限（北京）科技有限公司的注册商标。本文档中提及的其他公司、产品或服务名称可能是其他公司的商标或服务标志。

目录

1. HTML	4
1.1. HTML 简介、语义化标签	4
1.2. 浏览器加载、存储问题	9
1.3. 标签与浮动问题	10
1.4. 其他问题	11
2. CSS	13
2.1. CSS 盒子和选择器	13
2.2. CSS 布局	14
2.3. CSS3 问题	18
2.4. CSS 其他问题	20
3. JavaScript	27
3.1. JS 基础	27
3.2. JS 继承、原型链	30
3.3. JS_变量、作用域及内存	33
3.4. JS_函数与闭包	33
3.5. JS_DOM 与事件	34
3.6. JS 网络请求	35
3.7. JS 其他问题	36
3.8. JQuery	39
3.9. 其他框架	41
4. 总结部分	44
4.1. 知识点	44
4.2. 测试相关问题	46
4.3. 网络相关问题	47
4.4. 其他问题	54
面试注意点:	54

1. HTML

1.1. HTML 简介、语义化标签

1.1.1、Doctype作用？标准模式与兼容模式各有什么区别？

1) <!DOCTYPE>声明位于位于HTML文档中的第一行，处于 <html> 标签之前。告知浏览器的解析器用什么文档标准解析这个文档。DOCTYPE不存在或格式不正确会导致文档以兼容模式呈现。

2) 标准模式的排版 和JS运作模式都是以该浏览器支持的最高标准运行。在兼容模式中，页面以宽松的向后兼容的方式显示，模拟老式浏览器的行为以防止站点无法工作。

1.1.2、HTML5 为什么只需要写 <!DOCTYPE HTML>？

HTML5 不基于 SGML，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）；而 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。

1.1.3、如果我不放入<!DOCTYPE html>，HTML5 还会工作么？

不会，浏览器将不能识别他是 HTML 文档，同时 HTML5 的标签将不能正常工作

1.1.4、doctype（文档类型）的作用是什么？你知道多少种文档类型？

此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。该标签可声明三种 DTD 类型，分别表示严格版本、过渡版本以及基于框架的 HTML 文档。

1、HTML 4.01 规定了三种文档类型：Strict、Transitional 以及 Frameset。

2、XHTML 1.0 规定了三种 XML 文档类型：Strict、Transitional 以及 Frameset。

Standards（标准）模式（也就是严格呈现模式）用于呈现遵循最新标准的网页，而 Quirks（包容）模式（也就是松散呈现模式或者兼容模式）用于呈现为传统浏览器而设计的网页。

1.1.5、浏览器标准模式和怪异模式之间的区别是什么？

W3C 标准推出以后，浏览器都开始采纳新标准，但存在一个问题就是如何保证旧的网页还能继续浏览，在标准出来以前，很多页面都是根据旧的渲染方法编写的，如果用的标准来渲染，将导致页面显示异常。为保持浏览器渲染的兼容性，使以前的页面能够正常浏览，浏览器都保留了旧的渲染方法（如：微软的 IE）。这样浏览器渲染上就产生了 Quirks mode 和 Standards mode，两种渲染方法共存在一个浏览器上。IE 盒子模型和标准 W3C 盒子模型：ie 的 width 包括：padding\border。标准的 width 不包括：padding\border

在 js 中如何判断当前浏览器正在以何种方式解析？

document 对象有个属性 compatMode，它有两个值：BackCompat 对应 quirks mode，CSS1Compat 对应 strict mode。

1.1.6、什么是 HTML5？

H5=HTML5，指的是 HTML 超文本标记语言（HTML）的第五次重大修改，HTML 的第五代。它的主要目标是提供所有内容而不需要任何的像 flash，silverlight 等的额外插件，这些内容来自动画，视频，富 GUI 等。HTML5 是万维网联盟（W3C）和网络超文本应用技术工作组（WHATWG）之间合作输出的。

HTML5 具有的特点：

- (1)、语义化的标签：好处是网站加载更快；该标签举例：header、nav、article 等。
- (2)、增加了音频、视频标签：好处是取代 flash；该标签举例：audio、video。

(3)、canvas 标签：好处在浏览器上绘制图形或动画，取代 flash。

(4)、支持手机和平板的响应式布局。

HTML5 的缺点：低版本浏览器支持性不好，比如 IE9 以下的浏览器不支持 HTML5。

1.1.7、使用 XHTML 的局限有哪些？

XHTML 与 HTML 的区别为：

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。

局限：

所有的 XHTML 元素都必须被正确地嵌套，XHTML 必须拥有良好的结构，所有的标签必须小写，并且所有的 XHTML 元素必须被关闭。所有的 XHTML 文档必须拥有 DOCTYPE 声明，并且 html、head、title 和 body 元素必须存在。虽然代码更加的优雅，但缺少容错性，不利于快速开发。

1.1.8、如果把 HTML5 看作做一个开放平台，那它的构建模块有哪些？

<nav>,<header>,<section>,<footer>等

1.1.9、HTML 和 XHTML 有什么区别？

XHTML 全名是“可扩展超文本置标语言”(eXtensible HyperText Markup Language)，要说它和 HTML (HyperText Markup Language) 的区别就要从开始说起。

当初基于 HTML 衍生出 XML，因为 XML 拥有更加严谨的语法以及更好的语义环境，所以当年 W3C 的老大们纷纷认为 HTML 没用了，用 XML 更好。但是当时的环境直接让 WEB 跃进到 XML 实在有些不现实，毕竟当时还有许多书写不严谨的网页，而且应用于网页环境显然 HTML 更加有优势。于是讨论的结果是 XML 是未来的趋势，但是需要一个过程慢慢摸索实现的途径，所以就有了 XHTML，这是一个作为 HTML 到 XML 过渡过程的一个产物。

XHTML 是在 HTML 的基础上向 XML 过渡的东西，和 HTML 最大的区别就在于它更严谨而且语义化更强。比如 XHTML 规定所有标记都必须小写，因为 XML 需要区分大小写，而 HTML 不这么要求；XHTML 要求标签必须闭合，不允许开放标签存在比如“<p>第一段<p>第二段”这样的写法在 XHTML 里是错误的，但是 HTML 允许它的存在。目前 XHTML 的最新版本是 XHTML1.1，不过常用的仍是 XHTML1.0，XHTML1.1 规则太过严格，很多网站受不鸟（刚才提到的两个错误，在 XHTML1.0 里还是可以容忍，但是到了 XHTML1.1 就直接送你一个错误页面）。下一代的 XHTML 标准是 XHTML2.0，在 03 年开始制定，新出来的规范抛弃了大量 HTML 的属性和标记，而且允许使用 XML 的命名空间，而且开始允许扩展。总之说的很牛逼，不过 HTML5 兴起之后 XHTML2.0 的工作就被停止了，2006 年 W3C 关闭了 XHTML2.0 的工作组，转向 HTML5 规范的制定，未来 HTML5 普及以后估计 XHTML 就要成为历史了。

1.1.10、html5 有哪些新特性、移除了那些元素？如何处理 HTML5 新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？

HTML5 现在已经不是 SGML 的子集，主要是关于图像，位置，存储，多任务等功能的增加。

绘画 canvas;

用于媒介回放的 video 和 audio 元素;

本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失;

sessionStorage 的数据在浏览器关闭后自动删除;

语义化更好的内容元素，比如 article、footer、header、nav、section;

表单控件, calendar、date、time、email、url、search;
新的技术 webworker, websocket, Geolocation;
移除的元素:
纯表现的元素: basefont, big, center, font, s, strike, tt, u;
对可用性产生负面影响的元素: frame, frameset, noframes;
支持 HTML5 新标签:
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风格。
当然也可以直接使用成熟的框架、比如 html5shim;

1.1.11、如果页面使用 application/xhtml+xml 会有什么问题吗

一些老的浏览器不支持, 实际上, 任何最新的浏览器都将支持 application/xhtml+xml 媒体类型。
大多数浏览器也接受以 application/xml 发送的 XHTML 文档。

1.1.12、行内元素有哪些? 块级元素有哪些? 空(void)元素有那些?

首先: CSS 规范规定, 每个元素都有 display 属性, 确定该元素的类型, 每个元素都有默认的 display 值, 如 div 的 display 默认值为 “block”, 则为 “块级” 元素; span 默认 display 属性值为 “inline”, 是 “行内” 元素。

- (1) 行内元素有: a b span img input select strong (强调的语气)
- (2) 块级元素有: div ul ol li dl dt dd h1 h2 h3 h4...p
- (3) 常见的空元素:

 <hr> <input> <link> <meta>

鲜为人知的是:

<area> <base> <col> <command> <embed> <keygen> <param> <source> <track> <wbr>

1.1.13、页面导入样式时, 使用 link 和 @import 有什么区别?

- (1) link 属于 XHTML 标签, 除了加载 CSS 外, 还能用于定义 RSS, 定义 rel 连接属性等作用; 而 @import 是 CSS 提供的, 只能用于加载 CSS;
- (2) 页面被加载的时, link 会同时被加载, 而 @import 引用的 CSS 会等到页面被加载完再加载;
- (3) import 是 CSS2.1 提出的, 只在 IE5 以上才能被识别, 而 link 是 XHTML 标签, 无兼容问题;

1.1.14、介绍一下你对浏览器内核的理解?

主要分成两部分:

渲染引擎(layout engine 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。

1.1.15、常见的浏览器内核有哪些?

Trident 内核：IE,MaxThon,TT,The World,360,搜狗浏览器等。[又称 MSHTML]

Gecko 内核：Netscape6 及以上版本，FF,MozillaSuite/SeaMonkey 等

Presto 内核：Opera7 及以上。[Opera 内核原为：Presto，现为：Blink;]

Webkit 内核：Safari,Chrome 等。[Chrome 的：Blink (WebKit 的分支)]

1.1.16、如果网页内容需要支持多语言，你会怎么做

1) 采用统一编码 UTF-8 方式编码

对提供了多语言版本的网站来说，Unicode 字符集应该是最理想的选择。它是一种双字节编码机制的字符集，不管是东方文字还是西方文字，在 Unicode 中一律用两个字节来表示

2) 语言书写习惯&导航结构

3) 数据库驱动型网站 实现对不同语言数据信息的收集和检索。

1, 在数据库级别支撑多语言：为每种语言建立独立的数据库，不同语言的用户把持不同的数据库。

2, 在表级别支撑多语言：为每种语言建立独立的表，不同语言的用户把持不同的表，但是它们在同一个数据库中。

3, 在字段级别支撑多语言：在同一个表中为每种语言建立独立的字段，不同语言的用户把持不同的字段，它们在同一个表中。

4) 搜索引擎&市场推广

多语言网站实现计划：

1、静态：就是为每种语言分辨准备一套页面文件，要么通过文件后缀名来区分不同语言，要么通过子目录来区分不同语言。一旦用户选择了需要的语言后，主动跳转到相应的页面，首页以下其他链接也是按照同样方法处理。从保护的角度来看，通过子目录比通过文件后缀名来区分不同语言版本显得要简略明了。

2、动态：站点内所有页面文件都是动态页面文件（PHP，ASP 等）而不是静态页面文件，在需要输出语言文字的处所同一采用语言变量来表现，这些语言变量可以根据用户选择不同的语言赋予不同的值，从而能够实现在不同的语言环境下输出不同的文字。

1.1.17、简述一下你对 HTML 语义化的理解？

用正确的标签做正确的事情。

html 语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析；

即使在没有样式 CSS 情况下也以一种文档格式显示，并且是容易阅读的；

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，利于 SEO；

使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

1.1.18、请说出三种减少页面加载时间的方法。(加载时间指感知的时间或者实际加载时间)

1.优化图片

2.图像格式的选择（GIF：提供的颜色较少，可用在一些对颜色要求不高的地方）

3.优化 CSS（压缩合并 css，如 margin-top,margin-left...）

4.网址后加斜杠（如 www.campr.com/目录，会判断这个“目录是什么文件类型，或者是目录。）

5.标明高度和宽度（如果浏览器没有找到这两个参数，它需要一边下载图片一边计算大小，如果图片很多，浏览器需要不断地调整页面。这不但影响速度，也影响浏览体验。当浏览器知道了高度和宽度参数后，即使图片暂时无法显示，页面上也会腾出图片的空位，然后继续加载后面的内容。从而加载时间快了，浏览体验也更好了。）

6.减少 http 请求（合并文件，合并图片）。

1.1.19、请谈一下你对网页标准和标准制定机构重要性的理解。

w3c 存在的意义就是让浏览器兼容性问题尽量小，首先是他们对浏览器开发者的约束，然后是对开发者的约束。

1.1.20、使用 data- 属性的好处是什么

data-* 属性用于存储页面或应用程序的私有自定义数据。data-* 属性赋予我们在所有 HTML 元素上嵌入自定义 data 属性的能力。存储的（自定义）数据能够被页面的 JavaScript 中利用，以创建更好的用户体验（不进行 Ajax 调用或服务端数据库查询）。

data-* 属性包括两部分：

属性名不应该包含任何大写字母，并且在前缀 "data-" 之后必须有至少一个字符
属性值可以是任意字符串

1.1.21、H5 为什么这么火？H5 是哪一年产生的？H5 会火多久？

(1)、因为 H5 的后台很硬，是谷歌（google）和苹果（Apple）两大巨头公司，谷歌和苹果大力倡导 H5 的发展，也是 H5 的忠实的支持者和实践者（IOS+Android 系统占据手机市场 92.36%），H5 的产生主要目的也是干掉 flash，因为 flash 是 adobe 公司的，adobe 不是 W3C 组织的成员之一，苹果和谷歌等万维网联盟也不希望在自己的产品中使用加载量过大的 flash。

(2)、H5 是 2014 年 9 月份 W3C（万维网联盟，主要有苹果和谷歌公司）组织发布的。

(3)、刷脸时代（这里专指网站用户体验更加美观的时代）+移动端时代（手机+平板），会促使 H5 会在未来的有更加美好的发展前景，只要有网站，H5 就会很火，UI 也就很火，H5 的发展会让 UI 更加提高用户的满意度，H5 的火爆时代，会促进 UI 更好发展。

(4)、微信的发展，O2O 的促使 H5 更火。在这里不得不谈微信，如果没有微信，或许今天我们不会这样来讨论 H5，微信无意中养成了用户扫码的习惯，并通过公众号的内容生产及分发，以及微信本身已有的庞大的用户群体及社交属性，使基于网页的内容可以快速传播，真正带来了 APP 即浏览器的时代，取消了输入内容网址的麻烦。以微信为代表的超级社交 APP，解决了网页内容浏览及分发的通路。

H5 或许会没落，但会迎来 H6、H7...，只会变得越来越好，所以没必要担心 H5 会不会死，当下重在实践与积累，至少现在，Html5 的影响力，会超出你我的想象，也就意味着 H5 的好闺蜜 UI 会一直火下去。

1.1.22、div 是什么？在 div 出现之前做网站用什么布局？

div 是网站布局的盒子标签，div 出现是 table 布局，因为 table 布局嵌套很多，网站加载慢，布局层级不清晰。

1.1.23、html 是什么？css 是什么？js 是什么？

(1)、html 是超文本标记语言，他是做网站时候用的一些文本标记标签，比如 div、span 等。

(2)、css 是层叠样式表，是做网站的时候给标签来美化网站的样式，比如说 background（背景）、color（字体颜色）、height（高度）、width（宽度）等。

(3)、js=javascript 是网站中写前后台交互效果、网页动画效果的一种开发语言，比如鼠标点击事件（click）、前后台数据请求（ajax）等。

1.1.24、什么是静态网页？什么叫做动态网页？

(1)、静态网页：没有数据交互的网页，没有数据库参与，没有服务器端数据的加载。比如静态网页就是只有 html+css+JavaScript 做成的网站。

(2)、动态网页：有后台数据参与的网页，网页中的数据是从数据库取的，需要有后台逻辑的支持。比如动态网页就是 jsp 页面（后台语言是 java）、asp 页面（后台语言是 asp.net）。

1.2. 浏览器加载、存储问题

1.2.1、HTML5 的离线储存怎么使用，工作原理能不能解释一下？

在用户没有与因特网连接时，可以正常访问站点或应用，在用户与因特网连接时，更新用户机器上的缓存文件。

原理：HTML5 的离线存储是基于一个新建的.appcache 文件的缓存机制(不是存储技术)，通过这个文件上的解析清单离线

存储资源，这些资源就会像 cookie 一样被存储了下来。之后当网络在处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

如何使用：

- 1、页面头部像下面一样加入一个 manifest 的属性；
- 2、在 cache.manifest 文件的编写离线存储的资源；

CACHE MANIFEST

#v0.11

CACHE:

js/app.js

css/style.css

NETWORK:

resource/logo.png

FALLBACK:

/ /offline.html

- 3、在离线状态时，操作 window.applicationCache 进行需求实现。

1.2.2、浏览器是怎么对 HTML5 的离线储存资源进行管理和加载的呢？

1.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。

2.离线的情况下，浏览器就直接使用离线存储的资源。

1.2.3、请描述一下 cookies, sessionStorage 和 localStorage 的区别？

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），即会在浏览器和服务端间来回传递。

sessionStorage 和 localStorage 不会自动把数据发给服务器，仅在本地保存。sessionStorage 的概念很特别，引入了一个“浏览器窗口”的概念。sessionStorage 是在同源的同窗口（或 tab）中，始终存在的数据。也就是说只要这个浏览器窗口没有关闭，即使刷新页面或进入同源另一页面，数据仍然存在。关闭窗口后，sessionStorage 即被销毁。同时“独立”打开的不同窗口，即使是同一页面，sessionStorage 对象也是不同的 cookies 会发送到服务器端。

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。

有期时间：

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

1.2.4、如何实现浏览器内多个标签页之间的通信? (阿里)

WebSocket、SharedWorker；也可以调用 localStorage、cookies 等本地存储方式；
localStorage 另一个浏览上下文里被添加、修改或删除时，它都会触发一个事件，我们通过监听事件，控制它的值来进行页面信息通信；

注意 quirks: Safari 在无痕模式下设置 localStorage 值时会抛出 QuotaExceededError 的异常；

1.2.5、webSocket 如何兼容低浏览器? (阿里)

Adobe Flash Socket 、
ActiveX HTMLFile (IE) 、
基于 multipart 编码发送 XHR 、
基于长轮询的 XHR

1.3. 标签与浮动问题

1.3.1、iframe 有那些缺点?

iframe 会阻塞主页面的 Onload 事件；
搜索引擎的检索程序无法解读这种页面，不利于 SEO；
iframe 和主页面共享连接池，而浏览器对相同域的连接有限制，所以会影响页面的并行加载。

使用 iframe 之前需要考虑这两个缺点。如果需要使用 iframe，最好是通过 javascript 动态给 iframe 添加 src 属性值，这样可以绕开以上两个问题。

1.3.2、Label 的作用是什么? 是怎么用的?

label 标签来定义表单控制间的关系,当用户选择该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

```
<label for="Name">Number:</label>
<input type="text" name="Name" id="Name"/>
<label>Date:<input type="text" name="B"/></label>
```

1.3.3、HTML5 的 form 如何关闭自动完成功能?

给不想要提示的 form 或某个 input 设置为 autocomplete=off。

1.3.4、title 与 h1 的区别、b 与 strong 的区别、i 与 em 的区别?

title 属性没有明确意义只表示是个标题，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，使用阅读设备阅读网络时：会重读，而是展示强调内容。

i 内容展示为斜体，em 表示强调的文本；

Physical Style Elements -- 自然样式标签

b, i, u, s, pre

Semantic Style Elements -- 语义样式标签

strong, em, ins, del, code

应该准确使用语义样式标签，但不能滥用，如果不能确定时首选使用自然样式标签。

1.3.5、解释下浮动和它的工作原理。

关于浮动我们需要了解，浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。要想使元素浮动，必须为元素设置一个宽度（width）。虽然浮动元素不是文档流之中，但是它浮动后所处的位置依然是在浮动之前的水平方向上。由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样，下面的元素填补原来的位置。有些元素会在浮动元素的下方，但是这些元素的内容并不一定会被浮动的元素所遮盖，对内联元素进行定位时，这些元素会考虑浮动元素的边界，会围绕着浮动元素放置。也可以把浮动元素想象成是被块元素忽略的元素，而内联元素会关注浮动元素的。

1.3.6、列举不同的清除浮动的技巧，并指出它们各自适用的使用场景。

1.使用空标签清除浮动。这种方法是在所有浮动标签后面添加一个空标签定义 `css clear:both`。弊端就是增加了无意义标签。

2.使用 `overflow`。给包含浮动元素的父标签添加 `css 属性 overflow:auto;zoom:1;zoom:1` 用于兼容 IE6。

3.使用 `after` 伪对象清除浮动。该方法只适用于非 IE 浏览器。具体写法可参照以下示例。使用中需注意以下几点。

一、该方法中必须为需要清除浮动元素的伪对象中设置 `height:0`，否则该元素会比实际高出若干像素；

二、`content` 属性是必须的，但其值可以为空，`content` 属性的值设为“.”，空亦是可行的。

4.浮动外部元素

此三种方法各有利弊，使用时应择优选择，比较之下第二种方法更为可取。

1.4. 其他问题

1.4.1、解释下 CSS sprites，以及你要如何在页面或网站中使用它。

CSS Sprites 其实就是把网页中一些背景图片整合到一张图片文件中，再利用 CSS 的“`background-image`”，“`background-repeat`”，“`background-position`”的组合进行背景定位，`background-position` 可以用数字能精确的定位出背景图片的位置。

1.4.2、你熟悉 SVG 样式的书写吗？

SVG 指可伸缩矢量图形 (Scalable Vector Graphics)

SVG 用来定义用于网络的基于矢量的图形

SVG 使用 XML 格式定义图形

SVG 图像在放大或改变尺寸的情况下其图形质量不会有所损失

SVG 是万维网联盟的标准

SVG 与诸如 DOM 和 XSL 之类的 W3C 标准是一个整体

1.4.3、页面可见性 (Page Visibility API) 可以有哪些用途？

通过 `visibilityState` 的值检测页面当前是否可见，以及打开网页的时间等；在页面被切换到其他后台进程的时候，自动暂停音乐或视频的播放；

1.4.4、如何在页面上实现一个圆形的可点击区域？

- 1、`map+area` 或者 `svg`
- 2、`border-radius`

3、纯 js 实现 需要求一个点在不在圆上简单算法、获取鼠标坐标等等

1.4.5、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;overflow:hidden;background:red"></div>
```

1.4.6、网页验证码是干嘛的，是为了解决什么安全问题。

区分用户是计算机还是人的公共全自动程序。可以防止恶意破解密码、刷票、论坛灌水；有效防止黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试。

1.4.7、如何优化网页的打印样式？

```
<link rel = "stylesheet" type = "text/css" media = "screen" href = "xxx.css"/>
```

其中 media 指定的属性就是设备，显示器上就是 screen，打印机则是 print，电视是 tv，投影仪是 projection。

但打印样式表也应注意以下事项：

打印样式表中最好不要用背景图片，因为打印机不能打印 CSS 中的背景。如要显示图片，请使用 html 插入到页面中。

最好不要使用像素作为单位，因为打印样式表要打印出来的会是实物，所以建议使用 pt 和 cm。

隐藏掉不必要的内容。（@print div{display:none;}）

打印样式表中最好少用浮动属性，因为它们会消失。如果想要知道打印样式表的效果如何，直接在浏览器上选择打印预览就可以了。

1.4.8、你做的页面在哪些浏览器测试过？这些浏览器的内核分别是什么

ie(ie 内核) 火狐 (Gecko) 谷歌 (webkit,Blink) opera(Presto),Safari(wbkit)

1.4.9、什么是语义化的 HTML？

直观的认识标签 对于搜索引擎的抓取有好处，用正确的标签做正确的事情！html 语义化就是让页面的内容结构化，便于对浏览器、搜索引擎解析；在没有样式 CCS 情况下也以一种文档格式显示，并且是容易阅读的。搜索引擎的爬虫依赖于标记来确定上下文和各个关键字的权重，利于 SEO。使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

1.4.10、HTML5 为什么只需要写 !DOCTYPE HTML？

HTML5 不基于 SGML，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）；而 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。

1.4.11、Doctype 作用？标准模式与兼容模式各有什么区别？

!DOCTYPE 声明位于 HTML 文档中的第一行，处于 html 标签之前。告知浏览器的解析器用什么文档标准解析这个文档。DOCTYPE 不存在或格式不正确会导致文档以兼容模式呈现。

标准模式的排版 和 JS 运作模式都是以该浏览器支持的最高标准运行。在兼容模式中，页面以宽松的向后兼容的方式显示，模拟老式浏览器的行为以防止站点无法工作。

2. CSS

2.1. CSS 盒子和选择器

2.1.1、介绍一下标准的 CSS 的盒子模型？低版本 IE 的盒子模型有什么不同的？

- (1) 有两种， IE 盒子模型、W3C 盒子模型；
- (2) 盒模型： 内容(content)、填充(padding)、边界(margin)、 边框(border)；
- (3) 区别： IE 的 content 部分把 border 和 padding 计算了进去；

2.1.2、CSS 选择符有哪些？哪些属性可以继承？

- 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 ;

2.1.3、CSS 优先级算法如何计算？

优先级就近原则，同权重情况下样式定义最近者为准；

载入样式以最后载入的定位为准；

优先级为：

important > id > class > tag

important 比 内联优先级高

2.1.4、CSS3 新增伪类有那些？

举例：

- p:first-of-type 选择属于其父元素的首个 <p> 元素的每个 <p> 元素。
- p:last-of-type 选择属于其父元素的最后 <p> 元素的每个 <p> 元素。
- p:only-of-type 选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。
- p:only-child 选择属于其父元素的唯一子元素的每个 <p> 元素。
- p:nth-child(2) 选择属于其父元素的第二个子元素的每个 <p> 元素。

:after 在元素之前添加内容,也可以用来做清除浮动。

:before 在元素之后添加内容

:enabled

:disabled 控制表单控件的禁用状态。

:checked 单选框或复选框被选中。

2.1.5、display:inline-block 什么时候会显示间隙？(携程)

移除空格、使用 margin 负值、使用 font-size:0、letter-spacing、word-spacing

2.1.6、overflow: scroll 时不能平滑滚动的问题怎么处理？

- 1、阻止所有能导致页面滚动的事件。//scroll 不能阻止，只能阻止 mousewheel，鼠标拽滚动条就悲剧了；
- 2、bodyoverflow:hidden//win 下右侧滚动条会消失导致页面横移，移动端阻止不了；
- 3、把滚动部分单独放在一个 div 里，和弹出部分同级，body 和 window 同高。//所有涉及 offset/scrollTop 的方法都要修改。fix 并且 width100%的元素(比如微博顶栏)会压在内容区滚动条上；
- 4、弹出时算 scrollTop，给内容区 fix 然后 top 移动到目前位置，同时 body 给一个 overflow-y:scroll 强撑出滚动条。

2.1.7、有一个高度自适应的 div，里面有两个 div，一个高度 100px，希望另一个填满剩下的高度。

- 1、如果外层 div 高度自适应于内部，就完全不需要额外写规则了，另外一个 DIV 绝对能撑高外层 div，填得紧紧实实的。
- 2、如果是外层 div 自适应于它的父级，纯 CSS 的办法是有的。
- 3、absolute positioning
外层 position: relative;
百分百自适应元素直接 position: absolute; top: 100px; bottom: 0; left: 0

2.2. CSS 布局

2.2.1、如何居中 div？如何居中一个浮动元素？如何让绝对定位的 div 居中？

给 div 设置一个宽度，然后添加 margin:0 auto 属性

```
div{
  width:200px;
  margin:0 auto;
}
```

居中一个浮动元素

确定容器的宽高 宽 500 高 300 的层
设置层的外边距

相对定位的 div 居中：

```
.div {
  width:500px ; height:300px;//高度可以不设
  margin: -150px 0 0 -250px;
  position:relative;
  background-color:pink;    //方便看效果
  left:50%;
  top:50%;
}
```

让绝对定位的 div 居中：

```
.div {
  position: absolute;
  width: 1200px;
  background: none;
  margin: 0 auto;
  top: 0;
```

```
left: 0;
bottom: 0;
right: 0;
}
```

2.2.2、display 有哪些值？说明他们的作用。

block 像块类型元素一样显示。
 none 缺省值。像行内元素类型一样显示。
 inline-block 像行内元素一样显示，但其内容像块类型元素一样显示。
 list-item 像块类型元素一样显示，并添加样式列表标记。
 table 此元素会作为块级表格来显示
 inherit 规定应该从父元素继承 display 属性的值

2.2.3、position 的值 relative 和 absolute 定位原点是？

absolute
 生成绝对定位的元素，相对于值不为 static 的第一个父元素进行定位。
 fixed （老 IE 不支持）
 生成绝对定位的元素，相对于浏览器窗口进行定位。
 relative
 生成相对定位的元素，相对于其正常位置进行定位。
 static
 默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right z-index 声明）。
 inherit
 规定从父元素继承 position 属性的值。

2.2.4、li 与 li 之间有看不见的空白间隔是什么原因引起的？有什么解决办法？

行框的排列会受到中间空白（回车\空格）等的影响，因为空格也属于字符，这些空白也会被应用样式，占据空间，所以会有间隔，把字符大小设为 0，就没有空格了。

2.2.5、请解释一下为什么会出现浮动和什么时候需要清除浮动？清除浮动的方式

浮动 从何 而来 我们为何要清除浮动 清除浮动的原理是什么
<http://www.bitscn.com/school/HTMLCSS/201312/284207.html>

1) 普通流：很多人或者文章称之为文档流或者普通文档流，其实标准里根本就没有这个词。如果把文档流直译为英文就是 document flow，但标准里只有另一个词，叫做 普通流（normal flow），或者称之为常规流。

2) 浮动：浮动的框可以左右移动，直至它的外边缘遇到包含框或者另一个浮动框的边缘。浮动框不属于文档中的普通流，当一个元素浮动之后，不会影响到 块级框的布局而只会影响内联框（通常是文本）的排列，文档中的普通流就会表现得和浮动框不存在一样，当浮动框高度超出包含框的时候，也就会出现包含框不会自动伸高来闭合浮动元素。顾名思义，就是漂浮于普通流之上，像浮云一样，但是只能左右浮动。

正是因为浮动的这种特性，导致本属于普通流中的元素浮动之后，包含框内部由于不存在其他普通流元素了，也就表现出高度为 0（高度塌陷）。在实际布局中，往往这并不是我们所希望的，所以需要清除浮动的元素，使其包含框表现出正常的高度。

清除浮动的方式：

1.使用空标签清除浮动。这种方法是在所有浮动标签后面添加一个空标签定义 css clear:both. 弊端就是增加了无意义标签。

2.使用 `overflow`。给包含浮动元素的父标签添加 `css` 属性 `overflow:auto;zoom:1;zoom:1` 用于兼容 IE6。

3.使用 `after` 伪对象清除浮动。该方法只适用于非 IE 浏览器。具体写法可参照以下示例。使用中需注意以下几点。

一、该方法中必须为需要清除浮动元素的伪对象中设置 `height:0`，否则该元素会比实际高出若干像素；

二、`content` 属性是必须的，但其值可以为空，`content` 属性的值设为“.”，空亦是可以的。

4.浮动外部元素

此三种方法各有利弊，使用时应择优选择，比较之下第二种方法更为可取。

2.2.6、用纯 CSS 创建一个三角形的原理是什么？

把上、左、右三条边隐藏掉（颜色设为 `transparent`）

```
#demo {
  width: 0;
  height: 0;
  border-width: 20px;
  border-style: solid;
  border-color: transparent transparent red transparent;
}
```

2.2.7、一个满屏‘品’字布局如何设计？

简单的方式：

上面的 `div` 宽 100%，

下面的两个 `div` 分别宽 50%，

然后用 `float` 或者 `inline` 使其不换行即可

2.2.8、在网页中的应该使用奇数还是偶数的字体？为什么呢？

偶数字体：

一、UI 设计师的原因

多数设计师用的设计软件（如：`ps`）大多数都是偶数，所以前端工程师一般都是用偶数字体

二、浏览器的原因

其一是为了迁就 `ie6`，万恶的 `ie6` 会把定义为 `13px` 的字渲染成 `14px`，你可以写个页面试试

还有一个原因是，偶数宽的汉字，比如 `12px` 宽的汉字，去掉 1 像素的间距，填充了像素的实际宽是 `11px`，这样汉字的中竖线左右是平分的，以“中”这个字为例，在 12 像素时，竖线在中间，左右各 5 像素，显得均衡。

其二像谷歌一些比较流行的浏览器一般会有个默认的最小字体，而且对奇数字体渲染的不太好看

三、实际应用

偶数字号相对更容易和 `web` 设计的其他部分构成比例关系。比如：当我用了 `14 px` 的正文字号，我可能会在一些地方用 $14 \times 0.5 = 7 \text{ px}$ 的 `margin`，在另一些地方用 $14 \times 1.5 = 21 \text{ px}$ 的标题字号。`Windows` 自带的点阵宋体（中易宋体）从 `Vista` 开始只提供 `12`、`14`、`16 px` 这三个大小的点阵，而 `13`、`15`、`17 px` 时用的是小一号的点阵（即每个字占的空间大了 1 `px`，但点阵没变），于是略显稀疏。

2.2.9、margin 和 padding 分别适合什么场景使用

何时应当使用 `margin`：

需要在 border 外侧添加空白时。

空白处不需要背景（色）时。

上下相连的两个盒子之间的空白，需要相互抵消时。如 15px + 20px 的 margin，将得到 20px 的空白。

何时应当用 padding:

需要在 border 内测添加空白时。

空白处需要背景（色）时。

上下相连的两个盒子之间的空白，希望等于两者之和时。如 15px + 20px 的 padding，将得到 35px 的空白。

个人认为: margin 是用来隔开元素与元素的间距; padding 是用来隔开元素与内容的间隔。margin 用于布局分开元素使元素与元素互不相干; padding 用于元素与内容之间的间隔，让内容（文字）与（包裹）元素之间有一段“呼吸距离”。

2.2.10、全屏滚动的原理是什么？用到了 CSS 的那些属性

主要呈现方式有两种，一种是整体的元素一直排列下去，假设有五个需要展示的全屏页面，那么高度是 500%，只是展示 100%，剩下的可以通过 transform 进行 Y 轴定位，也可以通过 margin-top 实现，第二种就是所有的子元素和页面一样，都显示在当前页面，简单的实现第一种页面

2.2.11、CSS 隐藏元素的几种方法（至少说出三种）

Opacity:元素本身依然占据它自己的位置并对网页的布局起作用。它也将响应用户交互;

Visibility:与 opacity 唯一不同的是它不会响应任何用户交互。此外，元素在读屏软件中也会被隐藏;

Display:display 设为 none 任何对该元素直接打用户交互操作都不可能生效。此外，读屏软件也不会读到元素的内容。这种方式产生的效果就像元素完全不存在;

Position:不会影响布局，能让元素保持可以操作;

Clip-path:clip-path 属性还没有在 IE 或者 Edge 下被完全支持。如果要在你的 clip-path 中使用外部的 SVG 文件，浏览器支持度还要低;

2.2.12、CSS 居中（包括水平居中和垂直居中）

一、内联元素居中方案

水平居中设置:

1.行内元素 设置 text-align:center;

2.Flex 布局设置 display:flex;justify-content:center;(灵活运用,支持 Chrome, Firefox, IE9+)

垂直居中设置:

1.父元素高度确定的单行文本（内联元素）设置 height = line-height;

2.父元素高度确定的多行文本（内联元素）

a:插入 table （插入方法和水平居中一样），然后设置 vertical-align:middle;

b:先设置 display:table-cell 再设置 vertical-align:middle;

二、块级元素居中方案

水平居中设置:

1.定宽块状元素 设置 左右 margin 值为 auto;

2.不定宽块状元素

a:在元素外加入 table 标签（完整的，包括 table、tbody、tr、td），该元素写在 td 内，然后设置 margin 的值为 auto;

b:给该元素设置 display:inline 方法; c:父元素设置 position:relative 和 left:50%，子元素设置 position:relative 和 left:50%;

垂直居中设置:

使用 position:absolute (fixed),设置 left、top、margin-left、margin-top 的属性;

利用 position:fixed (absolute) 属性, margin:auto 这个必须不要忘记了;
利用 display:table-cell 属性使内容垂直居中;
使用 css3 的新属性 transform:translate(x,y)属性;
使用:before 元素;

2.2.13、写出几种 IE6 BUG 的解决方法

- 1.双边距 BUG float 引起的 使用 display
- 2.3 像素问题 使用 float 引起的 使用 display:inline -3px
- 3.超链接 hover 点击后失效 使用正确的书写顺序 link visited hover active
- 4.le z-index 问题 给父级添加 position:relative
- 5.Png 透明 使用 js 代码 改
- 6.Min-height 最小高度 ! Important 解决'
- 7.select 在 ie6 下遮盖 使用 iframe 嵌套
- 8.为什么没有办法定义 1px 左右的宽度容器 (IE6 默认的行高造成的, 使用 over:hidden,zoom:0.08 line-height:1px)

2.3. CSS3 问题

2.3.1、CSS3 有哪些新特性?

新增各种 CSS 选择器 (:not(input): 所有 class 不是 "input" 的节点)

圆角 (border-radius:8px)
多列布局 (multi-column layout)
阴影和反射 (Shadow\Reflect)
文字特效 (text-shadow、)
文字渲染 (Text-decoration)
线性渐变 (gradient)
旋转 (transform)

增加了旋转,缩放,定位,倾斜,动画, 多背景

transform:\scale(0.85,0.90)\ translate(0px,-30px)\ skew(-9deg,0deg)\Animation:

2.3.2、请解释一下 CSS3 的 Flexbox (弹性盒布局模型),以及适用场景?

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

2.3.3、::before 和 :after 中双冒号和单冒号 有什么区别？解释一下这 2 个伪元素的作用

单冒号(:)用于 CSS3 伪类，双冒号(::)用于 CSS3 伪元素。

伪元素由双冒号和伪元素名称组成。双冒号是在 css3 规范中引入的，用于区分伪类和伪元素。但是伪类兼容现存样式，浏览器需要同时支持旧的伪类，比如 :first-line、:first-letter、:before、:after 等。

对于 CSS2 之前已有的伪元素，比如:before，单冒号和双冒号的写法::before 作用是一样的。

提醒，如果你的网站只需要兼容 webkit、firefox、opera 等浏览器，建议对于伪元素采用双冒号的写法，如果不得不兼容 IE 浏览器，还是用 CSS2 的单冒号写法比较安全

::before 就是以子元素的存在，定义在元素主体内容之前的一个伪元素。但是并不存在于 dom 中，只是显示在页面上而已。同理，after 就是在主体内容之后显示的

2.3.4、设置元素浮动后，该元素的 display 值是多少？

浮动元素自动设置为"block"元素，而不管他们之前是什么。这说明浮动元素上的{display: inline;}会被忽略，事实上所有的浏览器没有呈现任何改变，包括 IE

2.3.5、font-style 属性可以让它赋值为 "oblique" oblique 是什么意思？

让一种字体标识为斜体(oblique)，如果没有这种格式，就使用 italic 字体。这里描述所用的 "oblique" 和 "italic" 都是倾斜的意思。"oblique" 在维基百科里的解释就是一种排版术语，就是一种倾斜的文字，但不是斜体。一种字库不能同时提供斜体和 oblique 两种字体，因为 oblique 基本上是一种模仿的斜体，而不是真正的斜体。所以，如果一种字库里没有提供斜体字，那当使用 CSS 的 font-style: italic 时，浏览器实际上是按 font-style: oblique 显示的。

2.3.6、position:fixed;在 android 下无效怎么处理

加入如下标记

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>
```

2.3.7、CSS 选择符有哪些？哪些属性可以继承？优先级算法如何计算？ CSS3 新增伪类有那些？

选择符:

id 选择器 (# myid)

类选择器 (.myclassname)

标签选择器 (div, h1, p)

相邻选择器 (h1 + p)

子选择器 (ul > li)

后代选择器 (li a)

通配符选择器 (*)

属性选择器 (a[rel = "external"])

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 ;

优先级算法:

优先级就近原则，同权重情况下样式定义最近者为准；
优先级为：

!important > id > class > tag

important 比 内联优先级高

CSS3 新增伪类：

p:first-of-type 选择属于其父元素的首个元素的每个元素。

p:last-of-type 选择属于其父元素的最后元素的每个元素。

p:only-of-type 选择属于其父元素唯一的元素的每个元素。

p:only-child 选择属于其父元素的唯一子元素的每个元素。

p:nth-child(2) 选择属于其父元素的第二个子元素的每个元素。

:enabled :disabled 控制表单控件的禁用状态。

:checked 单选框或复选框被选中。

2.4. CSS 其他问题

2.4.1、为什么要初始化 CSS 样式。

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。

当然，初始化样式会对 SEO 有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化。

最简单的初始化方法： * {padding: 0; margin: 0;} （强烈不建议）

淘宝的样式初始化代码：

```
body, h1, h2, h3, h4, h5, h6, hr, p, blockquote, dl, dt, dd, ul, ol, li, pre, form, fieldset, legend,
button, input, textarea, th, td { margin:0; padding:0; }
```

```
body, button, input, select, textarea { font:12px/1.5tahoma, arial, \5b8b\4f53; }
```

```
h1, h2, h3, h4, h5, h6 { font-size:100%; }
```

```
address, cite, dfn, em, var { font-style:normal; }
```

```
code, kbd, pre, samp { font-family:couriernew, courier, monospace; }
```

```
small { font-size:12px; }
```

```
ul, ol { list-style:none; }
```

```
a { text-decoration:none; }
```

```
a:hover { text-decoration:underline; }
```

```
sup { vertical-align:text-top; }
```

```
sub { vertical-align:text-bottom; }
```

```
legend { color:#000; }
```

```
fieldset, img { border:0; }
```

```
button, input, select, textarea { font-size:100%; }
```

```
table { border-collapse:collapse; border-spacing:0; }
```

2.4.2、absolute 的 containing block(容器块)计算方式跟正常流有什么不同？

无论属于哪种，都要先找到其祖先元素中最近的 position 值不为 static 的元素，然后再判断：

1、若此元素为 inline 元素，则 containing block 为能够包含这个元素生成的第一个和最后一个 inline box 的 padding box (除 margin, border 外的区域) 的最小矩形；

2、否则,则由这个祖先元素的 padding box 构成。

如果都找不到，则为 initial containing block。

补充:

1. static(默认的)/relative: 简单说就是它的父元素的内容框 (即去掉 padding 的部分)
2. absolute: 向上找最近的定位为 absolute/relative 的元素
3. fixed: 它的 containing block 一律为根元素(html/body), 根元素也是 initial containing block

2.4.3、CSS 里的 visibility 属性有个 collapse 属性值是干嘛用的? 在不同浏览器下以后什么区别?

当一个元素的 visibility 属性被设置成 collapse 值后, 对于一般的元素, 它的表现跟 hidden 是一样的。但例外的是, 如果这个元素是 table 相关的元素, 例如 table 行, table group, table 列, table column group, 它的表现却跟 display: none 一样, 也就是说, 它们占用的空间也会释放。

在谷歌浏览器里, 使用 collapse 值和使用 hidden 值没有什么区别。

在火狐浏览器、Opera 和 IE11 里, 使用 collapse 值的效果就如它的字面意思: table 的行会消失, 它的下面一行会补充它的位置。

2.4.4、经常遇到的浏览器的兼容性有哪些? 原因, 解决方法是什么, 常用 hack 的技巧?

png24 位的图片在 ie6 浏览器上出现背景, 解决方案是做成 PNG8。

浏览器默认的 margin 和 padding 不同。解决方案是加一个全局的{margin:0;padding:0;}来统一。

IE6 双边距 bug:块属性标签 float 后, 又有横行的 margin 情况下, 在 ie6 显示 margin 比设置的大。

浮动 ie 产生的双倍距离 #box{ float:left; width:10px; margin:0 0 0 100px;}这种情况之下 IE 会产生 20px 的距离, 解决方案是在 float 的标签样式控制中加入 ——_display:inline;将其转化为行内属性。(_这个符号只有 ie6 会识别)

渐进识别的方式, 从总体中逐渐排除局部。

首先, 巧妙的使用 “\9” 这一标记, 将 IE 浏览器从所有情况中分离出来。接着, 再次使用 “+” 将 IE8 和 IE7、IE6 分离开来, 这样 IE8 已经独立识别。

CSS

```
.bb{
    background-color:#f1ee18;/*所有识别*/
    .background-color:#00deff\9; /*IE6、7、8 识别*/
    +background-color:#a200ff;/*IE6、7 识别*/
    _background-color:#1e0bd1;/*IE6 识别*/
}
```

IE 下, 可以使用获取常规属性的方法来获取自定义属性, 也可以使用 getAttribute() 获取自定义属性; Firefox 下, 只能使用 getAttribute() 获取自定义属性。解决方法: 统一通过 getAttribute() 获取自定义属性。

IE 下, even 对象有 x, y 属性, 但是没有 pageX, pageY 属性;

Firefox 下, event 对象有 pageX, pageY 属性, 但是没有 x, y 属性。

解决方法: (条件注释) 缺点是在 IE 浏览器下可能会增加额外的 HTTP 请求数。

Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示, 可通过加入 CSS 属性 -webkit-text-size-adjust: none; 解决。

超链接访问过后 hover 样式就不出现了 被点击访问过的超链接样式不在具有 hover 和 active 了, 解决方法是改变 CSS 属性的排列顺序: L-V-H-A: a:link {} a:visited {} a:hover {} a:active {}

2.4.5、position 跟 display、margin collapse、overflow、float 这些特性相互叠加后会怎么样？

这个题目应该能拆成几个点来回答：

1、'display'、'position' 和 'float' 的相互关系；

1)、display 属性规定元素应该生成的框的类型。 block 象块类型元素一样显示，none 缺省值。象行内元素类型一样显示， inline-block 象行内元素一样显示，但其内容象块类型元素一样显示，list-item 象块类型元素一样显示，并添加样式列表标记

2)、position 属性规定元素的定位类型。 absolute 表示生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位；fixed（老 IE 不支持）生成绝对定位的元素，相对于浏览器窗口进行定位；relative 生成相对定位的元素，相对于其正常位置进行定位；static 默认值。没有定位，元素出现在正常的流中

3)、Float 也是一种布局方式，它定义元素在哪个方向浮动。在 CSS 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。在布局过程中也经常使用它来达到左右并排布局的效果

2、position 跟 display、overflow、float 下的 margin collapse。

把它看作是一个类似优先级的机制，"position:absolute" 和 "position:fixed" 优先级最高，有它存在的时候，浮动不起作用，'display' 的值也需要调整；其次，元素的 'float' 特性的值不是 "none" 的时候或者它是根元素的时候，调整 'display' 的值；最后，非根元素，并且非浮动元素，并且非绝对定位的元素，'display' 特性值同设置值。

2.4.6、对 BFC 规范(块级格式化上下文：block formatting context)的理解？

(W3C CSS 2.1 规范中的一个概念,它是一个独立容器,决定了元素如何对其内容进行定位,以及与其他元素的关系和相互作用。)一个页面是由很多个 Box 组成的,元素的类型和 display 属性,决定了这个 Box 的类型。

不同类型的 Box,会参与不同的 Formatting Context (决定如何渲染文档的容器),因此 Box 内的元素会以不同的方式渲染,也就是说 BFC 内部的元素和外部的元素不会互相影响。

2.4.7、css 定义的权重

以下是权重的规则：标签的权重为 1，class 的权重为 10，id 的权重为 100，以下例子是演示各种定义的权重值：

```
/*权重为 1*/
div{
}
/*权重为 10*/
.class1{
}
/*权重为 100*/
#id1{
}
/*权重为 100+1=101*/
#id1 div{
}
/*权重为 10+1=11*/
.class1 div{
}
/*权重为 10+10+1=21*/
.class1 .class2 div{
```


}

如果权重相同，则最后定义的样式会起作用，但是应该避免这种情况出现

2.4.8、移动端的布局用过媒体查询吗

css 的媒体查询允许通过 @media 标签为特定媒体的浏览器设定样式，其中包含众多筛选，功能强大。 @media (min-width: 700px) { ... }

2.4.9、CSS 优化、提高性能的方法有哪些

加载性能：比如不要用 import 啊，压缩啊等等，主要是从减少文件体积、减少阻塞加载、提高并发方面入手的，任何 hint 都逃不出这几个大方向。

渲染性能：渲染性能是 CSS 优化最重要的关注对象。页面渲染 junky 过多？看看是不是大量使用了 text-shadow？是不是开了字体抗锯齿？CSS 动画怎么实现的？合理利用 GPU 加速了吗？什么你用了 Flexible Box Model？有没有测试换个 layout 策略对 render performance 的影响？这个方面搜索一下 CSS render performance 或者 CSS animation performance 也会有一堆一堆的资料可供参考。

可维护性、健壮性：命名合理吗？结构层次设计是否足够健壮？对样式进行抽象复用了吗？优雅的 CSS 不仅仅会影响后期的维护成本，也会对加载性能等方面产生影响。这方面可以多找一些 OOCSS（不是说就要用 OOCSS，而是说多了解一下）等等不同 CSS Strategy 的信息，取长补短。

1、可重用性：一般来说，一个项目的整体设计风格是一致的，页面中肯定有几个风格一致但有些许不同的模块，如何在尽可能多地重用 CSS 代码，尽可能少地增加新代码，这是 CSS 代码中非常重要的一点。如果 CSS 代码的可重用性高，我们可能只需要写一些不一样的地方，对页面性能和可维护性、提高开发效率都有很大的帮助。

2、可扩展性：如果产品增加了某个功能，我们应该保证新增加的 CSS 代码不会影响到旧的 CSS 代码和页面，并且尽可能少地增加新代码而重用旧代码。

3、可修改性：如果某个模块产品经理觉得要修改样式，或者要删掉它，如果没有规划好相应的 CSS 代码，过了一段时间之后，开发人员可能已经不记得这段代码作用了几个地方，不敢修改或删除它，这样下去 CSS 代码也就越来越多，影响了页面的性能，还造成了代码的复杂度。

2.4.10、浏览器是怎样解析 CSS 选择器的？

浏览器先产生一个元素集合，这个集合往往由最后一个部分的索引产生（如果没有索引就是所有元素的集合）。然后向上匹配，如果不符合上一个部分，就把元素从集合中删除，直到真个选择器都匹配完，还在集合中的元素就匹配这个选择器了。

2.4.11、介绍一下 Sass 和 Less 是什么？它们有何区别？

Sass (Syntactically Awesome Stylesheets) 是一种动态样式语言，语法跟 css 一样(但多了些功能)，比 css 好写，而且更容易阅读。Sass 语法类似与 Haml，属于缩排语法 (makeup)，用意就是为了快速写 Html 和 Css。

Less 一种动态样式语言。将 CSS 赋予了动态语言的特性，如变量，继承，运算，函数。LESS 既可以在客户端上运行（支持 IE 6+，Webkit，Firefox），也可一在服务端运行（借助 Node.js）。

区别：

(1))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，也可以在开发环节使用 Less，然后编译成 Css 文件，直接放到项目中，

也有 Less.app、SimpleLess、CodeKit.app 这样的工具，也有在线编译地址。

(2)变量符不一样，less 是@，而 Scss 是\$，而且变量的作用域也不一样，后面会讲到。

(3)输出设置，Less 没有输出设置，Sass 提供 4 中输出选项：nested, compact, compressed 和 expanded。

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

2.4.12、如何修改 chrome 记住密码后自动填充表单的黄色背景

可以对 input:-webkit-autofill 使用足够大的纯色内阴影来覆盖 input 输入框的黄色背景

```
input:-webkit-autofill {
  -webkit-box-shadow: 0 0 0px 1000px white inset;
}
```

2.4.13、怎么让 Chrome 支持小于 12px 的文字

可以使用到 css3 里的一个属性：transform:scale()

属性介绍可以戳这里：<http://www.w3cthtml.com/css3/properties/2d-transform/transform.html>

这个属性前给-webkit-谷歌前缀，那么就可以控制字体的大小

<style>

```
p{font-size:10px;-webkit-transform:scale(0.8);}
```

/*这里的数字 0.8，是缩放比例，可以根据情况变化。*/

</style>

<p>中梦测试 10px</p>

2.4.14、让页面里的字体变清晰，变细用 CSS 怎么做？（-webkit-font-smoothing: antialiased;）

CSS3 里面加入了一个“-webkit-font-smoothing”属性。这个属性可以使页面上的字体抗锯齿,使用后字体看起来会更清晰舒服。加上之后就顿时感觉页面小清晰了。

2.4.15、png、jpg、gif 这些图片格式解释一下，分别什么时候用。有没有了解过 webp？

png:意为可移植网络图形格式(Portable Network Graphic Format)

动画 它不支持动画

无损耗性 png 是一种无损耗的图像格式，这也意味着你可以对 png 图片做任何操作也不会使 得图像质量产生损耗。这也使得 png 可以作为 jpeg 编辑的过渡格式

水平扫描 像 GIF 一样，png 也是水平扫描的，这意味着水平重复颜色比垂直重复颜色的图片更小

间隔渐进显示 它支持间隔渐进显示，但是会造成图片大小变得更大

JPEG 代表 Joint Photographic Experts Group（联合图像专家组），这种格式经常写成 JPG，JPG 图片的扩展名为 jpg。

JPG 最主要的优点是能支持上百万种颜色，从而可以用来表现照片。此外，由于 JPG 图片使用更有效的有损压缩算法，从而使文件长度更小，下载时间更短。有损压缩会放弃图像中的某些细节，以减少文件长度。它的压缩比相当高，使用专门的 JPG 压缩工具其压缩比可达 180: 1，而且图像质量从浏览角度来讲质量受损不会太大，这样就大大方便了网络传输和磁盘交换文件。JPG 较 GIF 更适合于照片，因为在照片中损失一些细节不像对艺术线条那么明显。另外，JPG 对照片的压缩比例更大，而最后的质量也更好。

GIF 意为 Graphics Interchange format（图形交换格式），GIF 图片的扩展名是 gif。现在所有的图形浏览器都支持 GIF 格式，而且有的图形浏览器只熟悉 GIF 格式。GIF 是一种索引颜色格式，在颜色数很少的情况下，产生的文件极小，它的优点主要有：

1.GIF 格式支持背景透明。GIF 图片假如背景色设置为透明，它将与浏览器背景相结

合，生成非矩形的图片。

2.GIF 格式支持动画。在 Flash 动画出现之前，GIF 动画可以说是网页中唯一的动画形式。GIF 格式可以将单帧的图象组合起来，然后轮流播放每一帧而成为动画。虽然并不是所有的图形浏览器都支持 GIF 动画，但是最新的图形浏览器都已经支持 GIF 动画。

3.GIF 格式支持图形渐进。渐进是指图片渐渐显示在屏幕上，渐进图片将比非渐进图片更快地出现在屏幕上，可以让访问者更快地知道图片的概貌。

4.GIF 格式支持无损压缩。无损压缩是不损失图片细节而压缩图片的有效方法，由于 GIF 格式采用无损压缩，所以它更适合于线条、图标和图纸。

5.水平扫描 Gif 是使用了一种叫作 LZW 的算法进行压缩的，当压缩 gif 的过程中，像素是由上到下水平压缩的，这也意味着同等条件下，横向的 gif 图片比竖向的 gif 图片更加小。例如 500*10 的图片比 10*500 的图片更加小

GIF 格式的缺点同样相当明显。索引颜色是历史遗留的产物，在 DOS 下的老游戏几乎无一例外的采用索引颜色，这种格式本来早就应该淘汰了。但是由于带宽的限制，GIF 从 DOS 时代红到了 Internet 时代。GIF 这种索引颜色格式最大的缺点就是它只有 256 种颜色，这对于照片质量的图片是显然不够的。

2.4.16、什么是 Cookie 隔离？（或者说：请求资源的时候不要让它带 cookie 怎么做）

如果静态文件都放在主域名下，那静态文件请求的时候都带有的 cookie 的数据提交给 server 的，非常浪费流量，所以不如隔离开。

因为 cookie 有域的限制，因此不能跨域提交请求，故使用非主要域名的时候，请求头中就不会带有 cookie 数据，这样可以降低请求头的大小，降低请求时间，从而达到降低整体请求延时的目的。

同时这种方式不会将 cookie 传入 Web Server，也减少了 Web Server 对 cookie 的处理分析环节，提高了 webserver 的 http 请求的解析速度。

2.4.17、什么是 CSS 预处理器 / 后处理器？

预处理器例如：LESS、Sass、Stylus，用来预编译 Sass 或 less，增强了 css 代码的复用性，还有层级、mixin、变量、循环、函数等，具有很方便的 UI 组件模块化开发能力，极大的提高工作效率。

后处理器例如：PostCSS，通常被视为在完成的样式表中根据 CSS 规范处理 CSS，让其更有效；目前最常做的是给 CSS 属性添加浏览器私有前缀，实现跨浏览器兼容性的问题。

2.4.18、style 标签写在 body 后与 body 前有什么区别？

2.4.19、如果需要手动写动画，你认为最小时间间隔是多久，为什么？（阿里）

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 $1/60 * 1000ms = 16.7ms$

2.4.20、抽离样式模块怎么写，说出思路，有无实践经验？[阿里航旅的面试题]

2.4.21、什么叫做响应式？

针对不同的屏幕显示不同的网页布局，比如说大屏幕手机、小屏幕手机、平板能够适应不同的屏幕来显示网站

2.4.22、针对不同屏幕的响应式，UI 设计师该做几套设计图？

只做一套，大多 UI 设计师会使用 iPhone5 的尺寸来做一套（640X1136）。那么这种可以适配响应式的各种屏幕的吗？这种在 HTML5 中通过代码中的媒体查询来实现。媒体查询怎

么写？媒体查询的具体写法如下：

```
@media screen and (max-width:640px) { //最大宽度是 640px 的屏幕宽度
    nav li { //nav 标签下的 li 标签设置样式
        display: inline; //转化为行元素
    }
}
```

3. JavaScript

3.1. JS 基础

3.1.1、介绍 js 的基本数据类型

Undefined、Null、Boolean、Number、String、Symbol(new in ECMAScript 2015)

3.1.2、介绍 js 有哪些内置对象？

Object 是 JavaScript 中所有对象的父对象

数据封装类对象：Object、Array、Boolean、Number 和 String

其他对象：Function、Arguments、Math、Date、RegExp、Error

3.1.3、说几条写 JavaScript 的基本规范？

- 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。

3.1.4、JavaScript 有几种类型的值？，你能画一下他们的内存图吗

栈：原始数据类型（Undefined，Null，Boolean，Number、String）

堆：引用数据类型（对象、数组和函数）

两种类型的区别是：存储位置不同；

原始数据类型直接存储在栈(stack)中的简单数据段，占据空间小、大小固定，属于被频繁使用数据，所以放入栈中存储；

引用数据类型存储在堆(heap)中的对象,占据空间大、大小不固定,如果存储在栈中，将会影响程序运行的性能；引用数据类型在栈中存储了指针，该指针指向堆中该实体的起始地址。当解释器寻找引用值时，会首先检索其在栈中的地址，取得地址后从堆中获得实体

3.1.5、null，undefined 的区别？

`null` 表示一个对象被定义了，值为“空值”；

`undefined` 表示不存在这个值。

`typeof undefined` `//"undefined"`

`undefined` :是一个表示“无”的原始值或者说表示“缺少值”，就是此处应该有一个值，但是还没有定义。当尝试读取时会返回 `undefined`；例如变量被声明了，但没有赋值时，就等于 `undefined`

`typeof null` `//"object"`

`null` : 是一个对象(空对象，没有任何属性和方法)；

例如作为函数的参数，表示该函数的参数不是对象；

注意：

在验证 `null` 时，一定要使用 `===`，因为 `==` 无法分别 `null` 和 `undefined`

参考 <http://www.ruanyifeng.com/blog/2014/03/undefined-vs-null.html>

3.1.6、javascript 代码中的"use strict";是什么意思？使用它区别是什么？

use strict 是一种 ECMAScript 5 添加的（严格）运行模式,这种模式使得 Javascript 在更严格的条件下运行,使 JS 编码更加规范化的模式,消除 Javascript 语法的一些不合理、不严谨之处,减少一些怪异行为。默认支持的糟糕特性都会被禁用,比如不能用 with,也不能在意外的情况下给全局变量赋值;

全局变量的显示声明,函数必须声明在顶层,不允许在非函数代码块内声明函数,arguments.callee 也不允许使用;消除代码运行的一些不安全之处,保证代码运行的安全,限制函数中的 arguments 修改,严格模式下的 eval 函数的行为和非严格模式的也不相同;

提高编译器效率,增加运行速度;

为未来新版本的 Javascript 标准化做铺垫。

3.1.7、如何判断一个对象是否属于某个类？

使用 instanceof (待完善)

```
if(a instanceof Person){
    alert('yes');
}
```

3.1.8、new 操作符具体干了什么呢？

- 1、创建一个空对象,并且 this 变量引用该对象,同时还继承了该函数的原型。
- 2、属性和方法被加入到 this 引用的对象中。
- 3、新创建的对象由 this 所引用,并且最后隐式的返回 this。

```
var obj = {};
obj.__proto__ = Base.prototype;
Base.call(obj);
```

3.1.9、用原生 JavaScript 的实现过什么功能吗？

原生 JavaScript 常用的正则表达式

原生 JavaScript 实现字符串长度截取

原生 JavaScript 获取域名主机

原生 JavaScript 清除空格

原生 JavaScript 替换全部

原生 JavaScript 转义 html 标签

原生 JavaScript 还原 html 标签

原生 JavaScript 时间日期格式转换

原生 JavaScript 判断是否为数字类型

原生 JavaScript 设置 cookie 值

原生 JavaScript 获取 cookie 值

原生 JavaScript 加入收藏夹

原生 JavaScript 设为首页

原生 JavaScript 判断 IE6

原生 JavaScript 加载样式文件

原生 JavaScript 返回脚本内容

原生 JavaScript 清除脚本内容

原生 JavaScript 动态加载脚本文件

原生 JavaScript 返回按 ID 检索的元素对象

原生 JavaScript 返回浏览器版本内容

原生 JavaScript 元素显示的通用方法

3.1.10、document.write 和 innerHTML 的区别

document.write 只能重绘整个页面

innerHTML 可以重绘页面的一部分

3.1.11、数组和对象有哪些原生方法，列举一下？

数组： join(): 将数组元素按指定的分隔符拼接成一字符串返回，默认分隔符为英文逗号

sort(fn): 数组排序,默认是按英文字母的 ASC 码升序排列，比如 apple 排在 orange 前面

pop(): 删除数组的最后一个元素，把数组长度减 1，并且返回它删除的元素的值。

push(n1,n2,n3,...nx):向数组末尾添加一个或多个元素，并返回添加后数组的长度

reverse(): 颠倒数组中元素的顺序

shift(): 删除数组的第一个元素，并返回它删除的那个元素

slice(start,end): 用于截取 start 和 end 之间的数组元素并存入新数组返回

unshift(element1,...,element):向数组开头添加一个或多个元素，并返回添加后的数组长度。

至少要传一个参数。

3.1.12、用 js 实现千位分隔符?(来源：前端农民工，提示：正则+replace)

```
function commafy(num) {
    num = num + "";
    var reg = /(?!d+)(d{3})/;
    if(reg.test(num)){
        num = num.replace(reg, '$1,$2');
    }
    return num;
}
```

3.1.13、如何测试前端代码么？知道 BDD, TDD, Unit Test 么？知道怎么测试你的前端工程么 (mocha, sinon, jasmine, qUnit..)?

3.1.14、前端 templating(Mustache, underscore, handlebars)是干嘛的，怎么用？

Handlebars 是一款很高效的模版引擎,提供语意化的模版语句,最大的兼容 Mustache 模版引擎,提供最大的 Mustache 模版引擎兼容,无需学习新语法即可使用;

3.1.15、简述一下 Handlebars 的基本用法？

3.1.16、简述一下 Handlebars 的对模板的基本处理流程， 如何编译的？如何缓存的？

3.1.17、Object.is() 与原来的比较操作符 “ === ”、“ == ” 的区别？

两等号判等，会在比较时进行类型转换；

三等号判等(判断严格)，比较时不进行隐式类型转换，(类型不同则会返回 false)；

Object.is 在三等号判等的基础上特别处理了 NaN、-0 和 +0，保证 -0 和 +0 不再相同，但 Object.is(NaN, NaN) 会返回 true.Object.is 应被认为有其特殊的用途，而不能用它认为它比其它的相等对比更宽松或严格。

3.1.18、请指出 JavaScript 宿主对象 (host objects) 和原生对象 (native objects) 的区别？

一、原生对象

ECMA-262 把原生对象 (native object) 定义为“独立于宿主环境的 ECMAScript 实现提供的对象”。

Object、Function、Array、String、Boolean、Number、Date、RegExp、Error、EvalError、RangeError、ReferenceError、SyntaxError、TypeError、URIError、ActiveXObject(服务器方面)、Enumerator(集合遍历类)、RegExp (正则表达式)

由此可以看出，简单来说，原生对象就是 ECMA-262 定义的类 (引用类型)。

二、宿主对象

何为“宿主对象”？ECMAScript 中的“宿主”当然就是我们网页的运行环境，即“操作系统”和“浏览器”。所有非原生对象都是宿主对象 (host object)，即由 ECMAScript 实现的宿主环境提供的对象。

所有的 BOM 和 DOM 对象都是宿主对象。因为其对于不同的“宿主”环境所展示的内容不同。其实说白了就是，ECMAScript 官方未定义的对象都属于宿主对象，因为其未定义的对象大多数是自己通过 ECMAScript 程序创建的对象。TMD DOM 是 W3C 标准 (是 HTML 文档对象模型的英文缩写，Document Object Model for HTML)。

HTML DOM 定义了用于 HTML 的一系列标准的对象，以及访问和处理 HTML 文档的标准方法。

通过 DOM，可以访问所有的 HTML 元素，连同它们所包含的文本和属性。可以对其中的内容进行修改和删除，同时也可以创建新的元素。

3.1.19、例举 3 种强制类型转换和 2 种隐式类型转换？

强制 (parseInt, parseFloat, number) 隐式 (== - ===);

3.2. JS 继承、原型链

3.2.1、JavaScript 原型，原型链？有什么特点？

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，也就是我们平时所说的原型链的概念。

关系：instance.constructor.prototype = instance.__proto__

特点：JavaScript 对象是通过引用来传递的，我们创建的每个新对象实体中并没有一份属于自己的原型副本。当我们修改原型时，与之相关的对象也会继承这一改变。

当我们需要一个属性的时，Javascript 引擎会先看当前对象中是否有这个属性，如果没有的话，就会查找他的 Prototype 对象是否有这个属性，如此递推下去，一直检索到 Object 内建对象。

```
function Func(){}
Func.prototype.name = "Sean";
Func.prototype.getInfo = function() {
    return this.name;
}
var person = new Func();//现在可以参考 var person = Object.create(oldObject);
console.log(person.getInfo());//它拥有了 Func 的属性和方法/"Sean"
console.log(Func.prototype);// Func { name="Sean", getInfo=function() }
```

3.2.2、Javascript 如何实现继承？

- 1、构造继承
- 2、原型继承
- 3、实例继承
- 4、拷贝继承

原型 prototype 机制或 apply 和 call 方法去实现较简单，建议使用构造函数与原型混合方式。

```
function Parent(){
    this.name = 'wang';
}
function Child(){
    this.age = 28;
}
Child.prototype = new Parent();//继承了 Parent，通过原型
var demo = new Child();
alert(demo.age);
alert(demo.name);//得到被继承的属性
```

3.2.3、JavaScript 继承的几种实现方式？

参考：构造函数的继承 <http://www.ruanyifeng.com/blog/2010/05/object-oriented-javascript-inheritance.html>,

非构造函数的继承 <http://www.ruanyifeng.com/blog/2010/05/object-oriented-javascript-inheritance-continued.html>;

3.2.4、javascript 创建对象的几种方式？

javascript 创建对象简单的说,无非就是使用内置对象或各种自定义对象，当然还可以用 JSON；但写法有很多种，也能混合使用。

- 1、对象字面量的方式

```
person={firstname:"Mark",lastname:"Yun",age:25,eyecolor:"black"};
```

- 2、用 function 来模拟无参的构造函数

```
function Person(){}
var person=new Person();//定义一个 function，如果使用 new"实例化",该 function
可以看作是一个 Class
```

```
person.name="Mark";
person.age="25";
person.work=function(){
    alert(person.name+" hello...");
}
```

```
person.work();
```

- 3、用 function 来模拟参构造函数来实现（用 this 关键字定义构造的上下文属性）

```
function Pet(name,age,hobby){
    this.name=name;//this 作用域：当前对象
    this.age=age;
    this.hobby=hobby;
    this.eat=function(){
```

```

        alert("我叫"+this.name+",我喜欢"+this.hobby+",是个程序员");
    }
}
var maidou =new Pet("麦兜",25,"coding");//实例化、创建对象
maidou.eat();//调用 eat 方法
4、用工厂方式来创建（内置对象）
var wcDog =new Object();
    wcDog.name="旺财";
    wcDog.age=3;
    wcDog.work=function(){
        alert("我是"+wcDog.name+",汪汪汪.....");
    }
    wcDog.work();
5、用原型方式来创建
function Dog(){
    Dog.prototype.name="旺财";
    Dog.prototype.eat=function(){
        alert(this.name+"是个吃货");
    }
    var wangcai =new Dog();
    wangcai.eat();
6、用混合方式来创建
function Car(name,price){
    this.name=name;
    this.price=price;
}
Car.prototype.sell=function(){
    alert("我是"+this.name+", 我现在卖"+this.price+"万元");
}
var camry =new Car("凯美瑞",27);
camry.sell();

```

3.2.5、Javascript 中，有一个函数，执行时对象查找时，永远不会去查找原型，这个函数是？

hasOwnProperty

JavaScript 中 hasOwnProperty 函数方法是返回一个布尔值，指出一个对象是否具有指定名称的属性。此方法无法检查该对象的原型链中是否具有该属性；该属性必须是对象本身的一个成员。

使用方法：

object.hasOwnProperty(propertyName)

其中参数 object 是必选项。一个对象的实例。

propertyName 是必选项。一个属性名称的字符串值。

如果 object 具有指定名称的属性，那么 JavaScript 中 hasOwnProperty 函数方法返回 true，反之则返回 false。

3.2.6、谈谈 This 对象的理解。

this 总是指向函数的直接调用者（而非间接调用者）；如果有 new 关键字，this 指向 new

出来的那个对象；在事件中，`this` 指向触发这个事件的对象，特殊的是，IE 中的 `attachEvent` 中的 `this` 总是指向全局对象 `Window`；

3.2.7、new 操作符具体干了什么呢？

- (1) 创建一个空对象，并且 `this` 变量引用该对象，同时还继承了该函数的原型。
- (2) 属性和方法被加入到 `this` 引用的对象中。
- (3) 新创建的对象由 `this` 所引用，并且最后隐式的返回 `this`。

3.3. JS_变量、作用域及内存

3.3.1、Javascript 作用链域？

全局函数无法查看局部函数的内部细节，但局部函数可以查看其上层的函数细节，直至全局细节。

当需要从局部函数查找某一属性或方法时，如果当前作用域没有找到，就会上溯到上层作用域查找，直至全局函数，这种组织形式就是作用域链。

3.3.2、JavaScript 中的作用域与变量声明提升？

javascript 的变量声明具有 `hoisting` 机制，JavaScript 引擎在执行的时候，会把所有变量的声明都提升到当前作用域的最前面。当前作用域内的声明都会提升到作用域的最前面，包括变量和函数的声明

3.4. JS_函数与闭包

3.4.1、什么是闭包（closure），为什么要用它？

闭包是指有权访问另一个函数作用域中变量的函数，创建闭包的最常见的方式就是在一个函数内创建另一个函数，通过另一个函数访问这个函数的局部变量，利用闭包可以突破作用域链，将函数内部的变量和方法传递到外部。

闭包的特性：

1. 函数内再嵌套函数
2. 内部函数可以引用外层的参数和变量
3. 参数和变量不会被垃圾回收机制回收

3.4.2、.call() 和 .apply() 的区别？

例子中用 `add` 来替换 `sub`，`add.call(sub,3,1) == add(3,1)`，所以运行结果为：`alert(4)`；
注意：js 中的函数其实是对象，函数名是对 `Function` 对象的引用。

```
function add(a,b)
{
    alert(a+b);
}
function sub(a,b)
{
```



```

    alert(a-b);
  }
  add.call(sub,3,1);
  Object.call(this,obj1,obj2,obj3)
  Object.apply(this,arguments)

```

3.4.3、闭包是什么，有什么特性，对页面有什么影响?简要介绍你理解的闭包

闭包就是能够读取其他函数内部变量的函数。

3.5. JS_DOM 与事件

3.5.1、事件是？IE 与火狐的事件机制有什么区别？ 如何阻止冒泡？

1. 我们在网页中的某个操作（有的操作对应多个事件）。例如：当我们点击一个按钮就会产生一个事件。是可以被 JavaScript 侦测到的行为。
2. 事件处理机制：IE 是事件冒泡、Firefox 同时支持两种事件模型，也就是：捕获型事件和冒泡型事件；
3. `ev.stopPropagation()`;（旧 ie 的方法 `ev.cancelBubble = true;`）

3.5.2、DOM 操作——怎样添加、移除、移动、复制、创建和查找节点？

（1）创建新节点

```

createDocumentFragment()    //创建一个 DOM 片段
createElement()             //创建一个具体的元素
createTextNode()             //创建一个文本节点

```

（2）添加、移除、替换、插入

```

appendChild()
removeChild()
replaceChild()
insertBefore() //在已有的子节点前插入一个新的子节点

```

（3）查找

```

getElementsByName() //通过标签名称
getElementsByName() //通过元素的 Name 属性的值(IE 容错能力较强，会得到一个数组，其中包括 id 等于 name 值的)
getElementById()    //通过元素 Id，唯一性

```

3.5.3、我们给一个 dom 同时绑定两个点击事件，一个用捕获，一个用冒泡。会执行几次事件，会先执行冒泡还是捕获？

1. 原生事件的发生顺序

一般来讲，当为一个 a 标签添加 click 事件以后，点击这个标签，会先执行绑定的事件、后跳转页面。一个 input 绑定 blur 事件以后，你在 input 里输入完内容，点击提交按钮，会先发生 blur 事件，后发生 click 事件。当然，这是一般来讲。我在一个 React 项目中曾经发生表单提交时，先发生 click 事件，blur 事件没有来得及发生，造成表单内容没有检验就提交到后台，原因我至今没有找到，解决办法是在 click 事件上加一个 50ms 的延迟。

2. 自定义事件

自定义事件的发生比较容易控制，你什么时候触发(`dispatchEvent/fireEvent`)它，它就什么时候发生。

3. 绑定多个事件，且由用户行为触发

绑定在被点击元素的事件是按照代码顺序发生，其他元素通过冒泡或者捕获“感知”的事件，按照 W3C 的标准，先发生捕获事件，后发生冒泡事件。所有事件的顺序是：其他元素捕获阶段事件 -> 本元素代码顺序事件 -> 其他元素冒泡阶段事件

3.6. JS 网络请求

3.6.1、JSON 的了解？

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
如：{"age": "12", "name": "back"}

JSON 字符串转换为 JSON 对象：

```
var obj = eval('(' + str + ')');
var obj = str.parseJSON();
var obj = JSON.parse(str);
```

JSON 对象转换为 JSON 字符串：

```
var last=obj.toJSONString();
var last=JSON.stringify(obj);
```

3.6.2、js 延迟加载的方式有哪些？

defer 和 async、动态创建 DOM 方式（用得最多）、按需异步载入 js

3.6.3、Ajax 是什么？如何创建一个 Ajax？

ajax 的全称：Asynchronous Javascript And XML。

异步传输+js+xml。

所谓异步，在这里简单地解释就是：向服务器发送请求的时候，我们不必等待结果，而是可以同时做其他的事情，等到有了结果它自己会根据设定进行后续操作，与此同时，页面是不会发生整页刷新的，提高了用户体验。

- (1)创建 XMLHttpRequest 对象,也就是创建一个异步调用对象
- (2)创建一个新的 HTTP 请求,并指定该 HTTP 请求的方法、URL 及验证信息
- (3)设置响应 HTTP 请求状态变化的函数
- (4)发送 HTTP 请求
- (5)获取异步调用返回的数据
- (6)使用 JavaScript 和 DOM 实现局部刷新

3.6.4、同步和异步的区别？

同步的概念应该是来自于 OS 中关于同步的概念:不同进程为协同完成某项工作而在先后次序上调整(通过阻塞,唤醒等方式).同步强调的是顺序性.谁先谁后.异步则不存在这种顺序性.

同步：浏览器访问服务器请求，用户看得到页面刷新，重新发请求，等请求完，页面刷新，新内容出现，用户看到新内容,进行下一步操作。

异步：浏览器访问服务器请求，用户正常操作，浏览器后端进行请求。等请求完，页面不刷新，新内容也会出现，用户看到新内容。

3.6.5、如何解决跨域问题?

jsonp、iframe、window.name、window.postMessage、服务器上设置代理页面

3.6.6、请解释 JavaScript 的同源策略 (same-origin policy)。

同源策略是浏览器上为安全性考虑实施的非常重要的安全策略。

何谓同源：URL 由协议、域名、端口和路径组成，如果两个 URL 的协议、域名和端口相同，则表示他们同源。

同源策略：浏览器的同源策略，限制了来自不同源的"document"或脚本，对当前"document"读取或设置某些属性。

从一个域上加载的脚本不允许访问另外一个域的文档属性。

Ajax 跨域：Ajax (XMLHttpRequest)请求受到同源策略的限制。Ajax 通过 XMLHttpRequest 能够与远程的服务器进行信息交互，另外 XMLHttpRequest 是一个纯粹的 Javascript 对象，这样的交互过程，是在后台进行的，用户不易察觉。因此，XMLHTTP 实际上已经突破了原有的 Javascript 的安全限制。

3.6.7、请解释同步 (synchronous) 和异步 (asynchronous) 函数的区别。

同步和异步通常用来形容一次方法调用。同步方法调用一旦开始，调用者必须等到方法调用返回后，才能继续后续的行为。异步方法调用更像一个消息传递，一旦开始，方法调用就会立即返回，调用者就可以继续后续的操作。而，异步方法通常会在另外一个线程中，“真实”地执行着。整个过程，不会阻碍调用者的工作。

3.6.8、ajax 请求的时候 get 和 post 方式的区别?

一个在 url 后面 一个放在虚拟载体里面

有大小限制

安全问题

应用不同 一个是论坛等只需要请求的，一个是类似修改密码的;

3.6.9、ajax 请求时，如何解释 json 数据

使用 eval parse,鉴于安全性考虑 使用 parse 更靠谱;

3.6.10、异步加载的方式有哪些?

(1) defer, 只支持 IE

(2) async:

(3) 创建 script, 插入到 DOM 中, 加载完毕后 callBack

3.7. JS 其他问题

3.7.1、页面编码和被请求的资源编码如果不一致如何处理?

假如本文件的编码是 gbk 或 gb2312 的。而引入的 js 编码为 utf-8 的，那就需要在引入的时候<script src="http://www.xxx.com/test.js" charset="utf-8"></script>同理，如果你的页面是 utf-8 的，引入的 js 是 gbk 的，那么就需要加上 charset="gbk"

3.7.2、模块化开发怎么做?

立即执行函数,不暴露私有成员

```
var module1 = (function(){
    var _count = 0;
    var m1 = function(){
        //...
    };
    var m2 = function(){
        //...
    };
    return {
        m1 : m1,
        m2 : m2
    };
})();
```

3.7.3、如何编写高性能的 Javascript?

- 1、尽量不要用 for-in 循环去访问数组，建议用 for 循环进行循环；
- 2、建议将对象进行缓存处理，特别是 DOM 访问是比较消耗资源的；
- 3、建议不要在函数内进行过深的嵌套判断；
- 4、避免循环引用，防止内存泄漏；
- 5、建议避免在函数内返回一个未声明的变量，会污染外部变量；
- 6、var 声明变量，建议写在多行

3.7.4、那些操作会造成内存泄漏?

3.7.5、知道什么是 webkit 么？知道怎么用浏览器的各种工具来调试和 debug 代码么？

WebKit 是一个开源的浏览器引擎，与之相对应的引擎有 Gecko（Mozilla Firefox 等使用）和 Trident（也称 MSHTML，IE 使用）。

用浏览器的各种工具来调试和 debug 代码：

1. debugger;我以前也说过，你可以在 JavaScript 代码中加入一句 debugger;来手工造成一个断点效果。需要带有条件的断点吗？你只需要用 if 语句包围它：if (somethingHappens) { debugger;}但要记住在程序发布前删掉它们。

2.设置在 DOM node 发生变化时触发断点。到时候你会发现 DOM 不受你的控制，自己会发生一些奇怪的变化，让你很难找出问题的根源。谷歌浏览器的开发工具里有一个超级好用的功能，专门可以对付这种情况，叫做“Break on...”，你在 DOM 节点上右键，就能看到这个菜单项。断点的触发条件可以设置成这个节点被删除、节点的属性有任何变化，或它的某个子节点有变化发生。

3. Ajax 断点

XHR 断点，或 Ajax 断点，就像它们的名字一样，可以让我们设置一个断点，在特点的 Ajax 调用发生时触发它们。当你在调试 Web 应用的网络传输时，这一招非常的有效

4. 移动设备模拟环境。谷歌浏览器里有一些非常有趣的模拟移动设备的工具，帮助我们调试程序在移动设备里的运行情况。找到它的方法是：按 F12，调出开发者工具，然后按 ESC 键(当前 tab 不能是 Console)，你就会看到第二层调试窗口出现，里面的 Emulation 标签页里有各种模拟设备可选。当然，这不会就变成了真正的 iPhone，只是模拟了 iPhone 的尺寸，触摸事件和浏览器 User Agent 值。

5. 使用 Audits 改进你的网站 YSlow 是一个非常棒的工具。谷歌浏览器的开发者工具里也有一个非常类似的工具，叫 Audits。它可快速的审计你的网站，给你提出非常实际有效的优

化你的网站的建议和方法。

3.7.6、检测浏览器版本有哪些方式？

功能检测、userAgent 特征检测

比如：navigator.userAgent

/"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_2) AppleWebKit/537.36

(KHTML, like Gecko) Chrome/41.0.2272.101 Safari/537.36"

3.7.7、What is a Polyfill?

polyfill 是“在旧版浏览器上复制标准 API 的 JavaScript 补充”，可以动态地加载 JavaScript 代码或库，在不支持这些标准 API 的浏览器中模拟它们。例如，geolocation（地理位置）polyfill 可以在 navigator 对象上添加全局的 geolocation 对象，还能添加 getCurrentPosition 函数以及“坐标”回调对象，所有这些都是 W3C 地理位置 API 定义的对象和函数。因为 polyfill 模拟标准 API，所以能够以一种面向所有浏览器未来的方式针对这些 API 进行开发，一旦对这些 API 的支持变成绝对大多数，则可以方便地去掉 polyfill，无需做任何额外工作。

3.7.8、做的项目中，有没有用过或自己实现一些 polyfill 方案（兼容性处理方案）？

比如：html5shiv、Geolocation、Placeholder

3.7.9、使用一种可以编译成 JavaScript 的语言来写 JavaScript 代码有哪些优缺点？

优点：编写的过程中的代码写法简洁，可读性好

缺点：

调试问题 由于是交叉编译，导致输出的代码很难被跟踪调试。

性能问题 由于 JavaScript 在浏览器中运行问题，本地代码编译 JavaScript 后的执行速度可能会下降。

3.7.10、请解释可变 (mutable) 和不变 (immutable) 对象的区别。

一个 immutable 对象（不可变对象）是指一旦创建之后状态不可改变的对象。mutable 对象（可变对象）是指创建之后也可以修改的对象。在有些情况下，对象也被认为是不可变的 (immutable)，即，一个对象包含的内部使用的属性改变了，但从外部看对象的状态并没有改变

不可变对象有几个优点：

线程安全

易于理解

比可变对象有更高的安全性

3.7.11、解释 function foo() {} 与 var foo = function() {} 用法的区别

方法一：foo 在进入函数执行上下文开始执行代码的时候，foo 是 undefined 的，知道执行完 var foo = function() {} 这句，foo 才会得到赋值；

方法二：在进入函数执行上下文开始执行代码的时候，foo 已经是一个 function 了。

3.7.12、请描述事件冒泡机制 (event bubbling)。

在一个对象上触发某类事件（比如单击 onclick 事件），如果此对象定义了此事件的处理程序，那么此事件就会调用这个处理程序，如果没有定义此事件处理程序或者事件返回 true，那么这个事件会向这个对象的父级对象传播，从里到外，直至它被处理（父级对象所有同类事件都将被激活），或者它到达了对象层次的最顶层，即 document 对象（有些浏览器是 window）。

3.7.13、请尽可能详尽的解释 Ajax 的工作原理。

创建 ajax 对象 (XMLHttpRequest/ActiveXObject(Microsoft.XMLHttp))

判断数据传输方式(GET/POST)

打开链接 open()

发送 send()

当 ajax 对象完成第四步 (onreadystatechange) 数据接收完成, 判断 http 响应状态 (status) 200-300 之间或者 304 (缓存) 执行回调函数

3.8. JQuery

3.8.1、JQuery 的源码看过吗? 能不能简单概况一下它的实现原理?

3.8.2、jQuery.fn 的 init 方法返回的 this 指的是什么对象? 为什么要返回 this?

3.8.3、jquery 中如何将数组转化为 json 字符串, 然后再转化回来?

jQuery 中没有提供这个功能, 所以你需要先编写两个 jQuery 的扩展:

```
$.fn.stringifyArray = function(array) {  
    return JSON.stringify(array)  
}  
$.fn.parseArray = function(array) {  
    return JSON.parse(array)  
}
```

然后调用:

```
$("").stringifyArray(array)
```

3.8.4、jQuery 的属性拷贝(extend)的实现原理是什么, 如何实现深拷贝?

3.8.5、jquery.extend 与 jquery.fn.extend 的区别?

(1). jQuery.extend(object); 它是为 jQuery 类添加类方法, 可以理解为添加静态方法。

(2). jQuery.fn.extend(object);

\$.fn 是指 jQuery 的命名空间, fn 上的成员(方法 function 及属性 property), 会对 jQuery 实例每一个有效。

3.8.6、谈一下 JQuery 中的 bind(),live(),delegate(),on()的区别?

bind(type,[data],fn) 为每个匹配元素的特定事件绑定事件处理函数

live(type,[data],fn) 给所有匹配的元素附加一个事件处理函数, 即使这个元素是以后再添加进来的

delegate(selector,[type],[data],fn) 指定的元素 (属于被选元素的子元素) 添加一个或多个事件处理程序, 并规定当这些事件发生时运行的函数

on(events,[selector],[data],fn) 在选择元素上绑定一个或多个事件的事件处理函数

差别:

.bind()是直接绑定在元素上

.live()则是通过冒泡的方式来绑定到元素上的。更适合列表类型的, 绑定到 document DOM 节点上。和.bind()的优势是支持动态数据。

.delegate()则是更精确的小范围使用事件代理, 性能优于.live()

.on()则是最新的 1.9 版本整合了之前的三种方式的新事件绑定机制

3.8.7、jQuery 一个对象可以同时绑定多个事件，这是如何实现的？

3.8.8、是否知道自定义事件。jQuery 里的 fire 函数是什么意思，什么时候用？

3.8.9、jQuery 是通过哪个方法和 Sizzle 选择器结合的？（jQuery.fn.find()进入 Sizzle）

3.8.10、jQuery 的队列是如何实现的？队列可以用在哪些地方？

3.8.11、针对 jQuery 性能的优化方法？

1、总是使用#id 去寻找 element.

2、在 Classes 前面使用 Tags

3、缓存 jQuery 对象

4、更好的利用链

5、使用子查询 jQuery 允许我们在一个包集上附加其它的选择器。因为我们已经在本地变量里保存了父对象这样会减少以后在选择器上的性能开销。

6、限制直接对 DOM 操作

7、事件委托（又名：冒泡事件）

8、消除查询浪费

9、遵从\$(window).load

10、压缩 JS

3.8.12、Jquery 与 jQuery UI 有啥区别？

jQuery 是一个 js 库，主要提供的功能是选择器，属性修改和事件绑定等等。

jQuery UI 则是在 jQuery 的基础上，利用 jQuery 的扩展性，设计的插件。提供了一些常用的界面元素，诸如对话框、拖动行为、改变大小行为等等

3.8.13、jQuery 和 Zepto 的区别？各自的使用场景？

1、Zepto 是为现代智能手机浏览器推出的 Javascript 框架，有和 jQuery 相似的语法，和 jQuery 相比下来，他有很多优点，大小方面，压缩后的 zepto.min.js 大小只有 21K，使用服务器端 gzip 压缩后大小只有 5~10K，可以说非常的小，功能很齐全，多出来了一些触摸屏的事件，它对 PC 的浏览器就不是那么理想，尤其是万恶 IE 直接过滤不兼容，用它来开发手机 iPhone 和 Android 网页是不错的选择。

2、jquery 主要是用在 PC 的网页中了，jquery 是目前最流行的 javascript 框架，以其兼容全部主流浏览器，插件丰富，代码简洁，最关键的是更新快，最好的 dom 选择器而被广泛被使用。而手机 wap 方面的话，jquery 也推出 jquery-mobile，也很好用。

简单的理解就是：

Zepto 手机专用

jQuery 电脑专用（但他又开发出了 jquery-mobile,专门用于手机）

3.8.14、针对 jQuery 的优化方法？

基于 Class 的选择性的性能相对于 Id 选择器开销很大，因为需遍历所有 DOM 元素。

频繁操作的 DOM，先缓存起来再操作。用 JQuery 的链式调用更好。

比如：var str=\$(“a”).attr(“href”);

for (var i = size; i < arr.length; i++) {}

for 循环每一次循环都查找了数组 (arr) 的.length 属性, 在开始循环的时候设置一个变量来存储这个数字, 可以让循环跑得更快:

```
for (var i = size, length = arr.length; i < length; i++) {}
```

3.8.15、jQueryUI 如何自定义组件?

jQuery 的 slideUp 动画, 如果目标元素是被外部事件驱动, 当鼠标快速地连续触发外部元素事件, 动画会滞后的反复执行, 该如何处理呢?

jQuery 中 slideUp、slideDown、animate 等动画运用时, 如果目标元素是被外部事件驱动, 当鼠标快速地连续触发外部元素事件, 动画会滞后的反复执行, 其表现不雅。则解决办法:

1、在触发元素上的事件设置为延迟处理, 即可避免滞后反复执行的问题 (使用 setTimeout)

2、在触发元素的事件时预先停止所有的动画, 再执行相应的动画事件 (使用 stop)

jquery stop: //语法结构

```
$("#div").stop();//停止当前动画, 继续下一个动画
```

```
$("#div").stop(true);//清除元素的所有动画
```

```
$("#div").stop(false, true);//让当前动画直接到达末状态, 继续下一个动画
```

```
$("#div").stop(true, true);//清除元素的所有动画, 让当前动画直接到达末状态
```

本人一般推荐使用第二种方法

```
$("#div").stop().animate({width:"100px"},100);
```

3.9. 其他框架

3.9.1、Node.js 的适用场景?

1. RESTful API 这是 NodeJS 最理想的应用场景, 可以处理数万条连接, 本身没有太多的逻辑, 只需要请求 API, 组织数据进行返回即可。它本质上只是从某个数据库中查找一些值并将它们组成一个响应。由于响应是少量文本, 入站请求也是少量的文本, 因此流量不高, 一台机器甚至也可以处理最繁忙的公司的 API 需求。

2. 统一 Web 应用的 UI 层。目前 MVC 的架构, 在某种意义上来说, Web 开发有两个 UI 层, 一个是在浏览器里面我们最终看到的, 另一个在 server 端, 负责生成和拼接页面。不论这种架构是好是坏, 但是有另外一种实践, 面向服务的架构, 更好的做前后端的依赖分离。如果所有的关键业务逻辑都封装成 REST 调用, 就意味着在上层只需要考虑如何用这些 REST 接口构建具体的应用。那些后端程序员们根本不操心具体数据是如何从一个页面传递到另一个页面的, 他们也不用管用户数据更新是通过 Ajax 异步获取的还是通过刷新页面。

3. 大量 Ajax 请求的应用。例如个性化应用, 每个用户看到的页面都不一样, 缓存失效, 需要在页面加载的时候发起 Ajax 请求, NodeJS 能响应大量的并发请求。总而言之, NodeJS 适合运用在高并发、I/O 密集、少量业务逻辑的场景。

3.9.2、(如果会用 node)知道 route, middleware, cluster, nodemon, pm2, server-side rendering 么?

3.9.3、解释一下 Backbone 的 MVC 实现方式?

3.9.4、什么是“前端路由”?什么时候适合使用“前端路由”?“前端路由”有哪些优点和缺点?

1, 什么是前端路由?

路由是根据不同的 url 地址展示不同的内容或页面。前端路由就是把不同路由对应不同的内容或页面的任务交给前端来做, 之前是通过服务端根据 url 的不同返回不同的页面实

现的。

2, 什么时候使用前端路由?

在单页面应用, 大部分页面结构不变, 只改变部分内容的使用

3, 前端路由有什么优点和缺点?

优点

用户体验好, 不需要每次都从服务器全部获取, 快速展现给用户

缺点

使用浏览器的前进, 后退键的时候会重新发送请求, 没有合理地利用缓存

单页面无法记住之前滚动的位置, 无法在前进, 后退的时候记住滚动的位置

3.9.5、react-router 路由系统的实现原理?

实现 URL 与 UI 界面的同步。其中在 react-router 中, URL 对应 Location 对象, 而 UI 是由 react components 来决定的, 这样就转变成 location 与 components 之间的同步问题。

3.9.6、对 Node 的优点和缺点提出了自己的看法?

(优点)因为 Node 是基于事件驱动和无阻塞的, 所以非常适合处理并发请求, 因此构建在 Node 上的代理服务器相比其他技术实现 (如 Ruby) 的服务器表现要好得多。此外, 与 Node 代理服务器交互的客户端代码是由 javascript 语言编写的, 因此客户端和服务端都用同一种语言编写, 这是非常美妙的事情。

(缺点) Node 是一个相对新的开源项目, 所以不太稳定, 它总是一直在变, 而且缺少足够多的第三方库支持。看起来, 就像是 Ruby/Rails 当年的样子。

3.9.7、你有哪些性能优化的方法?

(1) 减少 http 请求次数: CSS Sprites, JS、CSS 源码压缩、图片大小控制合适; 网页 Gzip, CDN 托管, data 缓存, 图片服务器。

(2) 前端模板 JS+数据, 减少由于 HTML 标签导致的带宽浪费, 前端用变量保存 AJAX 请求结果, 每次操作本地变量, 不用请求, 减少请求次数

(3) 用 innerHTML 代替 DOM 操作, 减少 DOM 操作次数, 优化 javascript 性能。

(4) 当需要设置的样式很多时设置 className 而不是直接操作 style。

(5) 少用全局变量、缓存 DOM 节点查找的结果。减少 IO 读取操作。

(6) 避免使用 CSS Expression (css 表达式) 又称 Dynamic properties(动态属性)。

(7) 图片预加载, 将样式表放在顶部, 将脚本放在底部 加上时间戳。

(8) 避免在页面的主体布局中使用 table, table 要等其中的内容完全下载之后才会显示出来, 显示 div+css 布局慢。对普通的网站有一个统一的思路, 就是尽量向前端优化、减少数据库操作、减少磁盘 IO。向前端优化指的是, 在不影响功能和体验的情况下, 能在浏览器执行的不要在服务端执行, 能在缓存服务器上直接返回的不要到应用服务器, 程序能直接取得的结果不要到外部取得, 本机内能取得的数据不要到远程取, 内存能取到的不要到磁盘取, 缓存中有的不要去数据库查询。减少数据库操作指减少更新次数、缓存结果减少查询次数、将数据库执行的操作尽可能的让你的程序完成 (例如 join 查询), 减少磁盘 IO 指尽量不使用文件系统作为缓存、减少读写文件次数等。程序优化永远要优化慢的部分, 换语言是无法“优化”的。

3.9.8、http 状态码有那些? 分别代表是什么意思?

100-199 用于指定客户端应相应的某些动作。

200-299 用于表示请求成功。

300-399 用于已经移动的文件并且常被包含在定位头信息中指定新的地址信息。

400-499 用于指出客户端的错误。400 1、语义有误, 当前请求无法被服务器理解。401

当前请求需要用户验证 403 服务器已经理解请求，但是拒绝执行它。

500-599 用于支持服务器错误。 503 - 服务不可用

3.9.9、一个页面从输入 URL 到页面加载显示完成，这个过程中都发生了什么？（流程说的越详细越好）

查找浏览器缓存

DNS 解析、查找该域名对应的 IP 地址、重定向（301）、发出第二个 GET 请求

进行 HTTP 协议会话

客户端发送报头(请求报头)

文档开始下载

文档树建立，根据标记请求所需指定 MIME 类型的文件

文件显示

浏览器这边做的工作大致分为以下几步：

加载：根据请求的 URL 进行域名解析，向服务器发起请求，接收文件（HTML、JS、CSS、图像等）。

解析：对加载到的资源（HTML、JS、CSS 等）进行语法解析，建议相应的内部数据结构（比如 HTML 的 DOM 树，JS 的（对象）属性表，CSS 的样式规则等等）

4. 总结部分

4.1. 知识点

4.1.1、前端开发涉及的知识点

HTML&CSS

对 Web 标准的理解、浏览器内核差异、兼容性、hack、CSS 基本功：布局、盒子模型、选择器优先级、HTML5、CSS3、Flexbox

JavaScript:

数据类型、运算、对象、Function、继承、闭包、作用域、原型链、事件、RegExp、JSON、Ajax、DOM、BOM、内存泄漏、跨域、异步装载、模板引擎、前端 MVC、路由、模块化、Canvas、ECMAScript、Nodejs

其他:

移动端、响应式、自动化构建、HTTP、离线存储、WEB 安全、优化、重构、团队协作、可维护、易用性、SEO、UED、架构、职业生涯、快速学习能力

4.1.2、作为一名前端工程师，无论工作年头长短都应该掌握的知识点:

- 1.HTML 与 XHTML —— 二者有什么区别，你觉得应该使用哪一个并说出理由。
- 2.盒模型 —— 外边距、内边距和边框之间的关系，及 IE8 以下版本的浏览器中的盒模型
- 3.严格模式与混杂模式 —— 如何触发这两种模式，区分它们有何意义。
- 4.块级元素与行内元素 —— 怎么用 CSS 控制它们、以及如何合理的使用它们
- 5.浮动元素 —— 怎么使用它们、它们有什么问题以及怎么解决这些问题。
- 6.DOM 结构 —— 两个节点之间可能存在哪些关系以及如何在节点之间任意移动。
- 7.DOM 操作 —— 如何添加、移除、移动、复制、创建和查找节点等。
- 8.事件 —— 如何使用事件，以及 IE 和标准 DOM 事件模型之间存在的差别。
- 9.JSON —— 作用、用途、设计结构。
- 10.XMLHttpRequest —— 这是什么、怎样完整地执行一次 GET 请求、怎样检测错误。

4.1.3、什么是前端工程师？什么是后端工程师？

前端工程师就是指的做静态网页的工程师：

- (1)、广义的前端分为三种：安卓工程师、ios 工程师、web 前端工程师。
- (2)、狭义的前端指的是 web 前端工程师，web 前端工程师指的是做静态的 PC 端和手机端静态网页的工程师。

4.1.4、前端语言有哪些？后端语言有哪些？

- (1)、前端语言：HTML、css、javascript。
- (2)、后端语言（服务器端语言）：php、java、asp.net。最近新出的 node.js

4.1.5、做一个网站的团队都需要哪些人？

- (1)、产品经理：设计这个产品，通常就是了解用户的网站需求，画原型图。
- (2)、项目经理：通常是对整个产品有一个整理管理和负责，通常是会代码技术的人来构建整个网站的代码框架，以后网站实现的全面管理。
- (3)、UI 设计师，通过原型图画 psd 设计图的。
- (4)、前端工程师，根据设计图来做静态网页，可能是原生 app 的 IOS 和安卓工程师，或者 web 端的 web 前端工程师。

(5)、后端工程师，通常就是做 java、asp.net、php 的工程师来写后端逻辑的工程师。

4.1.6、做一个网页设计师或者前端工程师，平常访问学习的 IT 网站都有哪些？

- (1)、W3C shool
- (2)、segmentFault
- (3)、csdn
- (4)、知乎论坛
- (5)、博客园

4.1.7、切图工作是 UI 设计师来做？还是前端工程师来做？

对于 app 工程师，也就是 ios 和 Android 工程师，大多由 UI 设计师来完成切图。

对于 web 前端工程师，也就是 PC 端、浏览端，大多有 web 前端工程师自己完成切图。

4.1.8、切图工程师、前端工程师、UI 设计师、美工、网页设计师区别是什么？

(1)、UI 设计师俗称美工，不过 UI 设计师工作高端、名字大气、工资上档次，不过大多公司都称呼 UI 为美工，你也不要介意的，不管他们怎么称呼的，反正就是做网站设计图的就 OK，别人怎么称呼不重要的了，只要你拿了高工资就是 UI 设计师了：

UI 的主要任务是设计。了解用户的意图，分析网站配色，基本布局。绘制出一个网站效果图。UI 需要掌握的知识体系应该包括网页设计，UI (User Interface) 用户界面人机交互、操作逻辑、界面美观的整体设计，UED (user experience design) 用户体验设计--简单来说就是如何使得网站更加便于交互。

(2)、前端开发：美工在完成设计效果图之后，由前端开发人员将其制作成为适合浏览器查看的 HTML 页面。由于现在移动互联网的大规模流行，加上各个不同厂商的浏览器的激烈竞争，前端开发的主要任务简单来说就是使网页在不同浏览器不同分辨率不同设备上提供相似或相近的浏览体验。前端开发需要掌握的知识体系主要是兼容性问题的解决，流畅完美的交互体验。具体到技术细节上就是 HTML，CSS，JavaScript，各大公司各种不同内核的浏览器、各种各样的 JS 库、简单的与后台交互的知识。

(3)、后台开发：前台开发完成之后，就是后台程序员的工作了，相比较前端来说，后台更像传统意义上的程序员。后台的工作简单来说就是网页文件对数据库的增删改查。

后台需要掌握的知识体系应该包括，编程基础，基本 HTML 语言，至少一门主流网页语言 (C#，C++，JAVA，PHP 等)，数据库的操作等等。

(4)、UI 设计师和网页设计师有什么区别？

其实网页设计，分出来有两块，一个是 UI 设计，一个是 web 前端。UI 设计自然要懂的更多的是 PS，FW，AI，CD 等制图软件，还有一些比较优秀的网页设计理念，切图等相关知识。

web 前端，需要的 html+css+javascript，通过这三个东西把设计图转换成代码。这一步所实现的就是设计图的静态化，也就是变成了网页形式。

网页设计师，是个很泛的概念，不过一般指的会偏重 UI 设计。稍微关注过网页设计领域的公司，有点规模的，招人都会写得比较详细，例如招 UI 设计师，或者招 web 前端工程师。

你找工作的时候，都找这些名称比较规范的，因为起码他们会区别职位的不同。一定要找的时候看清楚是 UI 设计师，不是找前端或者美工的，前端更多的写较高级的代码的，会比设计懂的知识更多的，美工其实就是淘宝装修店铺或者说简单的 PS 照片的，就是会玩美图秀秀或者 PS 简单的绘图就可以的了，UI 设计师才是咱们的 IT 行业“高大上”职位。

(5)、UI 设计师：“UI”的本义是用户界面，是英文 User 和 interface 的缩写。UI 设计师简称 UID (User Interface Designer)，指从事对软件的人机交互、操作逻辑、界面美观的整体设计工作的人。UI 设计师的涉及范围包括商用平面设计、高级网页设计、移动应用界面设计及部分包装设计，是目前中国信息产业中最为抢手的人才之一。UI 设计师的特点是：工资高、发展前景好，会一些简单的前端知识代码，做手机端和 PC 端的网站设计图。

4.1.9、UI 设计师的工作内容是什么？

- (1)、负责软件界面的美术设计、创意工作和制作工作；
- (2)、根据各种相关软件的用户群，提出构思新颖、有高度吸引力的创意设计；
- (3)、对页面进行优化，使用户操作更趋于人性化；
- (4)、维护现有的应用产品；
- (5)、收集和分析用户对于 GUI 的需求。

4.1.10、UI 设计师需要会使用的工具的简称都有哪些？

以下一些工具的简称，大家应该有一定的了解，以免被问到咱们只知道简称。

- (1)、AI (adobe illustrator)基于矢量的图形制作软件
- (2)、PS(adobe Photoshop)图像处理软件。
- (3)、DW(Adobe Dreamweaver)网页编辑器
- (4)、AE(Adobe After Effects)一款图形视频处理软件
- (5)、flash (Adobe Flash) 二维动画软件
- (6)、Axure (Axure RP) 快速原型设计工具
- (7)、墨刀 (MockingBot) 移动端原型工具
- (8)、Fireworks 网页作图软件

4.1.11、什么时网页三剑客？

网页三剑客，是一套强大的网页编辑工具，最初是由 Macromedia 公司开发出来的。由 Dreamweaver, Fireworks, Flash 三个软件组成，俗称网页三剑客。

Dreamweaver 是一个“所见即所得”的可视化网站开发工具，主要用于动态网页和静态网页的开发

Fireworks 主要是用于对网页上常用的 jpg、gif 的制作和处理，也可用于制作网页布局；

Flash 主要用来制作动画，现推出 Flash 平台，可预见有极好的前景。

4.2. 测试相关问题

4.2.1、对代码进行测试的有什么优缺点？

4.2.2、你会用什么工具测试你的代码功能？

一、QUnit 前端测试工具 QUnit 是一个强大的 JavaScript 单元测试框架，该框架是由 jQuery 团队的成员所开发，并且是 jQuery 的官方测试套件。Qunit 是 JQuery 的单元测试框架，并且被广泛使用在各个项目中。为了使用 Qunit，你仅仅需要去包含 2 个 Qunit 文件在你的 Html 页面。Qunit 包含了 qunit.js 作为运行器和测试框架，和 qunit.css 文件，座位测试套件页面显示测试结果样式。

二、Selenium 前端测试工具 Selenium 是 ThoughtWorks 专门为 Web 应用程序编写的一个验收测试工具。Selenium 也是一个用于 Web 应用程序测试的工具。Selenium 测试直接运行在浏览器中，就像真正的用户在操作一样。这个工具的主要功能包括：

1、测试与浏览器的兼容性——测试你的应用程序看是否能够很好得工作在不同浏览器和操作系统之上。

2、测试系统功能——创建回归测试检验软件功能和用户需求。

3、支持自动录制动作和自动生成 .Net、Java、Perl 等不同语言的测试脚本。

三、Jshint 前端测试工具

JSHint 跟 JSLint 非常像，都是一个 JavaScript 的代码质量检查工具，主要用来检查代码质量以及找出一些潜在的代码缺陷并提供相关的代码改进意见。

使用方式：

第一种方法：进入 JSHint 首页，粘贴你的代码，选择相关的选项，然后点击右下角的 Lint 按钮就可以了。

第二种方法：使用 Grunt 整合的 JSHint。

第三种方法：直接使用 API。

四、WebDriverTest 前端测试工具

WebDriver 就是基于 Selenium 的一个自动化测试类库，但它不再是运行在浏览器内的 JS 程序，而是自己可以控制浏览器。旨在改进 Selenium 中出现的诸多问题，并且提供了非常易用、可读性很强的 API。

4.2.3、单元测试与功能/集成测试的区别是什么？

单元测试：是指对软件中的最小可测试单元进行检查和验证

集成测试：也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求（如根据结构图）组装成为子系统或系统，进行集成测试。

系统测试：将已经集成好的软件系统，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合在一起，在实际使用环境下，对计算机系统进行一系列的组装测试和确认测试的工作。

根据不同的测试阶段，测试可以分为单元测试、集成测试、系统测试和验收测试。体现了测试由小到大、由内至外、循序渐进的测试过程。

单元测试的粒度最小。

集成测试界于单元测试和系统测试之间，起到“桥梁作用”。

系统测试的粒度最大，主要测试系统是否符合“需求规格说明书”。系统测试同时也是在经过以上各阶段测试确认之后，把系统完整地模拟客户环境来进行的测试。

4.2.4、代码风格 linting 工具的作用是什么？

4.2.4、你做的页面在哪些浏览器测试过？

IE 内核浏览器：360，傲游，搜狗，世界之窗，腾讯 TT。

非 IE 内核浏览器：firefox opera safari chrome

4.3. 网络相关问题

4.3.1、为什么传统上利用多个域名来提供网站资源会更有效？

除了节约 cookie 带宽的因素，另一个重要因素是节约主域名的连接数，从而提高客户端网络带宽的利用率，优化页面响应。因为老的浏览器（IE6 是典型），针对同一个域名只允许同时保持两个 HTTP 连接。将图片等资源请求分配到其他域名上，避免了大图片之类的并不一定重要的内容阻塞住主域名上其他后续资源的连接（比如 ajax 请求）

4.3.2、请尽可能完整得描述从输入 URL 到整个网页加载完毕及显示在屏幕上的整个流程。

- 1、输入地址
- 2、浏览器查找域名的 IP 地址。这一步包括 DNS 具体的查找过程，包括：浏览器缓存 -> 系统缓存 -> 路由器缓存...
- 3、浏览器向 web 服务器发送一个 HTTP 请求
- 4 服务器的永久重定向响应（从 http://example.com 到 http://www.example.com）
- 5、浏览器跟踪重定向地址

- 6、服务器处理请求
- 7、服务器返回一个 HTTP 响应
- 8、浏览器显示 HTML
- 9、浏览器发送请求获取嵌入在 HTML 中的资源（如图片、音频、视频、CSS、JS 等等）
- 10、浏览器发送异步请求

参考：<http://igoro.com/archive/what-really-happens-when-you-navigate-to-a-url/>

4.3.3、Long-Polling、Websockets 和 Server-Sent Event 之间有什么区别？ Long-Polling :

客户端使用普通的 http 方式向服务器端请求网页客户端执行网页中的 JavaScript 脚本，向服务器发送数据、请求信息服务器并不是立即就对客户端的请求作出响应，而是等待有效的更新当信息是有效的更新时，服务器才会把数据推送给客户端。当客户端接收到服务器的通知时，立即会发送一个新的请求，进入到下一次的轮询

Server Sent Events (SSE) / EventSource:客户端使用普通的 http 方式向服务器端请求网页。客户端执行网页中的 JavaScript 脚本，与服务器之间建立了一个连接当服务器端有有效的更新时，会发送一个事件到客户端服务器到客户端数据的实时推送，大多数内容是你需要的你需要一台可以做 Event Loop 的服务器不允许跨域的连接。

Websockets:客户端使用普通的 http 方式向服务器端请求网页客户端执行网页中的 JavaScript 脚本，与服务器之间建立了一个连接服务器和客户端之间，可以双向的发送有效数据到对方服务器可以实时的发送数据到客户端，同时客户端也可以实时的发送数据到服务器。你需要一台可以做 Event Loop 的服务器使用 WebSockets 允许跨域的建立连接，它同样支持第三方的 websocket 主机服务器，例如 Pusher 或者其它。这样你只需要关心客户端的实现，降低了开发难度。

4.3.4、请描述以下 request 和 response headers:

Diff. between Expires, Date, Age and If-Modified-...

Do Not Track、Cache-Control、Transfer-Encoding、ETag

X-Frame-Options

4.3.5、什么是 HTTP method? 请罗列出你所知道的所有 HTTP method，并给出解释

HTTP/1.1 协议中共定义了八种方法（有时也叫“动作”）来表明 Request-URI 指定的资源的不同操作方式

GET、POST、HEAD、OPTIONS、PUT、DELETE 和 TRACE

GET 向特定的资源发出请求。注意：GET 方法不应当被用于产生“副作用”的操作中，例如在 web app.中。其中一个原因是 GET 可能会被网络蜘蛛等随意访问。

POST 向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST 请求可能会导致新的资源的建立和/或已有资源的修改。

HEAD 向服务器索要与 GET 请求相一致的响应，只不过响应体将不会被返回。这一方法可以在不必传输整个响应内容的情况下，就可以获取包含在响应消息头中的元信息。

OPTIONS 返回服务器针对特定资源所支持的 HTTP 请求方法。也可以利用向 Web 服务器发送 '*' 的请求来测试服务器的功能性。

PUT 向指定资源位置上传其最新内容。

DELETE 请求服务器删除 Request-URI 所标识的资源。

TRACE 回显服务器收到的请求，主要用于测试或诊断。

4.3.6、请用自己的语言简述：XSS 是什么（举例说明）如何防治 XSS

1.xss (cross site scripting) 即跨站脚本。比如在有表单输入的地方，用户输入了类似

`<script>alert("ok")</script>`的东西，而服务器没有经过过滤就保存下来了，别的用户看到网页就会弹出提示框。当然，xss 攻击还可以做其他更加复杂的事情。比如，获取 cookie 什么的。

2.把"<"和">"转换为 html 字符实体可以防范大多数 xss 攻击。可以使用 js 过滤，也可以在服务器端过滤。

4.3.7、请用自己的话简述 什么是「闭包」。「闭包」的作用是什么。

“官方”的解释是：所谓“闭包”，指的是一个拥有许多变量和绑定了这些变量的环境的表达式（通常是一个函数），因而这些变量也是该表达式的一部分。

「函数」和「函数内部能访问到的变量」（也叫环境）的总和，就是一个闭包。

作用：简而言之，就是在函数执行完并返回后，闭包使得 Javascript 的垃圾回收机制 GC 不会收回函数所占用的资源，因为函数的内部函数的执行需要依赖函数中的变量。这是对闭包作用的描述，不专业也不严谨，但大概意思就是这样。

4.3.8、为什么 CSS 选择器的名称不要以 ad 开头？

其实是因为 Adblock、Adblock Plus 等浏览器插件会自动屏蔽一些以 ad 开头的标签和文件。

因为很多浏览器厂商或去广告插件，都会过滤 html 中含有“ad、advertisement”等字段的节点，如果使用“ad、advertisement”开头或者包含的 id、class 节点都会被 remove，所以不建议使用。

4.3.9、Fetch API 是什么？能代替 AJAX 吗？

1.Fetch 是浏览器提供的原生 AJAX 接口。使用 `window.fetch` 函数可以代替以前的 `$.ajax`、`$.get` 和 `$.post`。Fetch API 就是浏览器提供的用来代替 `jQuery.ajax` 的工具。

2.随着 React.js、Angular.js 和 Vue.js 这些前端框架的流行，很多单页面应用已经不再使用 jQuery 了，这意味着你要自己对 XMLHttpRequest 进行封装，而很多人选择封装一个跟 `jQuery.ajax` 差不多的接口。

3.Fetch API 的出现，就是为了给类似的操作流程定一个接口规范。换句话说，就是浏览器帮你把 `jQuery.ajax` 给实现了，以后大家都是用 `fetch` 来发送异步请求就好了。Fetch API 提供的一组对象 `window.fetch` 函数只是 Fetch API 提供的众多接口中的一个，还有很多有用的对象：

`window.Headers`
`window.Response`
`window.Request`

Fetch API 的特点

基于 Promise（如果你没有学过 Promise，强烈建议你学一学）
不需要依赖第三方库，就可以优雅地使用 AJAX

4.3.10、如何在不刷新页面的情况下改变 URL

需求整理如下：

点击分页页码可实现无刷新页面加载
同时 URL 在数据加载后会发生变化展示对应页码
刷新页面（带页码参数）会定位到当前页码
当点击返回时可会到上个页面

实现：

对于第 1 条，我们可以使用 `ajax` 动态获取对应页码的数据。

对于第 2 条，我们可以使用 html5 的 api 「history.pushState」，用于改变 URL。

对于第 3 条，我们可以根据 URL 中页码参数获取对应页码的数据再做展示。

对于第 4 条，可以使用 「window.onpopstate」来监听返回事件那 history.pushState 如何使用呢？比如当用户点击页码按钮时，可使用 ajax 获取对应页码的数据，拼装 DOM 放到页面上，然后调用下面的 setUrl 方法实现浏览器 URL 的更新。

```
function setUrl(page){
    var url = location.pathname + '?page=' + page
    history.pushState({url: url, title: document.title}, document.title, url)
}
```

history.pushState() 带有三个参数：一个状态对象，一个标题（现在被忽略了），以及一个可选的 URL 地址。

state object — 状态对象是一个由 pushState()方法创建的与历史记录相关的 JS 对象。

title — 火狐浏览器现在已经忽略此参数，将来也许可能被使用。考虑到将来有可能的改变，传递一个空字符串是安全的做法。当然，你可以传递一个短标题给你要转变成的状态。

URL — 这个参数提供了新历史记录的地址。请注意，浏览器在调用 pushState()方法后不会去加载这个 URL，但有可能在之后会这样做，比如用户重启浏览器之后。新的 URL 可以是绝对地址，也可以是相对地址。新 URL 必须和当前 URL 在同一个源下。

4.3.11、JS 里基本类型（值）和复杂类型（引用）有什么区别？

1. 基本类型变量存的是值，复杂类型的变量存的是内存地址。
2. 基本类型在赋值的时候拷贝值，复杂类型在赋值的时候只拷贝地址，不拷贝值。

```
var a = 1.23
```

如果计算机要存储一个整数，它可以用 32 个位（bit）来存储，一个小数，可以用 64 位来存储。但不管怎样，位数都是固定的。

```
对象类型的变量 var obj = {};var c = 1.628
```

为了解决 obj 存储的问题，计算机将程序里的内存分成两种，一种是上图所示，按顺序使用内存，每个数据占据的位数是固定的，这种内存叫做「栈内存」；另一个，就是专门用来存储位数不固定的数据，存的时候不一样按顺序一个一个存，这种内存叫做「堆内存」。

存储方式：

obj 在栈内存那里，只占固定位数（32 位或 64 位或其他都可以），里面存的并不是数据 {a:1.23,b:2.34}，里面存的是数据「在内存中的位置」（类似于引用或者指针）。堆内存里，会开辟一块空间来存放 {a:1.23, b:2.34}。

4.3.12、简述 Cookie 是什么？

1. Cookie 是浏览器访问服务器后，服务器传给浏览器的一段数据。
2. 浏览器需要保存这段数据，不得轻易删除。
3. 此后每次浏览器访问该服务器，都必须带上这段数据。Cookie 就是这么简单，这就是 Web 开发里 Cookie 的含义。

作用：

第一个作用是识别用户身份。

比如用户 A 用浏览器访问了 http://a.com，那么 http://a.com 的服务器就会立刻给 A 返回一段数据「uid=1」（这就是 Cookie）。当 A 再次访问 http://a.com 的其他页面时，就会附上「uid=1」这段数据。

同理，用户 B 用浏览器访问 http://a.com 时，http://a.com 发现 B 没有附带 uid 数据，就给 B 分配了一个新的 uid，为 2，然后返回给 B 一段数据「uid=2」。B 之后访问 http://a.com 的时候，就会一直带上「uid=2」这段数据。

借此，`http://a.com` 的服务器就能区分 A 和 B 两个用户了。

第二个作用是记录历史。

假设 `http://a.com` 是一个购物网站，当 A 在上面将商品 A1、A2 加入购物车时，JS 可以改写 Cookie，改为「`uid=1; cart=A1,A2`」，表示购物车里有 A1 和 A2 两样商品了。

这样一来，当用户关闭网页，过三天再打开网页的时候，依然可以看到 A1、A2 躺在购物车里，因为浏览器并不会无缘无故地删除这个 Cookie。

借此，就达到记录用户操作历史的目的了。

（上面的例子只是为了让大家了解 Cookie 的作用而构想出来的，实际的网站使用 Cookie 时会更谨慎一些。）

4.3.13、cookie、session、localStorage 分别是什么？有什么作用？

一、cookie

1.cookie 是存储在浏览器上的一小段数据，用来记录某些当页面关闭或者刷新后仍然需要记录的信息。在控制台用「`document.cookie`」查看你当前正在浏览的网站的 cookie。

2.cookie 可以使用 js 在浏览器直接设置（用于记录不敏感信息，如用户名），也可以在服务端通使用 HTTP 协议规定的 `set-cookie` 来让浏览器种下 cookie，这是最常见的做法。

（打开一个网站，清除全部 cookie，然后刷新页面，在 network 的 Response headers 试试找一找 `set-cookie` 吧）

3.每次网络请求 Request headers 中都会带上 cookie。所以如果 cookie 太多太大对传输效率会有影响。

4.一般浏览器存储 cookie 最大容量为 4k，所以大量数据不要存到 cookie

5.设置 cookie 时的参数：

(1).path: 表示 cookie 影响到的路径，匹配该路径才发送这个 cookie。expires 和 maxAge: 告诉浏览器 cookie 时候过期，maxAge 是 cookie 多久后过期的相对时间。不设置这两个选项时会产生 session cookie，session cookie 是 transient 的，当用户关闭浏览器时，就被清除。一般用来保存 session 的 session_id。

(2).secure: 当 secure 值为 true 时，cookie 在 HTTP 中是无效，在 HTTPS 中才有效

(3).httpOnly: 浏览器不允许脚本操作 `document.cookie` 去更改 cookie。一般情况下都应该设置这个为 true，这样可以避免被 xss 攻击拿到 cookie。[cookie 参数][简述 Cookie 是什么]

二、session

1.当一个用户打开淘宝登录后，刷新浏览器仍然展示登录状态。服务器如何分辨这次发起请求的用户是刚才登录过的用户呢？这里就使用了 session 保存状态。用户在输入用户名密码提交给服务端，服务端验证通过后会创建一个 session 用于记录用户的相关信息，这个 session 可保存在服务器内存中，也可保存在数据库中。

2.创建 session 后，会把关联的 session_id 通过 `setCookie` 添加到 http 响应头部中。

3.浏览器在加载页面时发现响应头部有 `set-cookie` 字段，就把这个 cookie 种到浏览器指定域名下。

4.当下次刷新页面时，发送的请求会带上这条 cookie，服务端在接收到后根据这个 session_id 来识别用户。

5.cookie 是存储在浏览器里的一小段「数据」，而 session 是一种让服务器能识别某个用户的「机制」，session 在实现的过程中需要使用 cookie。二者不是同一维度的东西。

三、localStorage

1.localStorage HTML5 本地存储 web storage 特性的 API 之一，用于将大量数据（最大 5M）

保存在浏览器中，保存后数据永远存在不会失效过期，除非用 js 手动清除。

2. 不参与网络传输。

3. 一般用于性能优化，可以保存图片、js、css、html 模板、大量数据。

4.3.14、为什么不建议将 font-size 设置为 12px 以下？

因为 Chrome 这款任性的浏览器做了如下限制：

1. font-size 有一个最小值 12px（不同操作系统、不同语言可能限制不一样），低于 12px 的，一律按 12px 显示。理由是 Chrome 认为低于 12px 的中文对人类的不友好的。

2. 但是允许你把 font-size 设置为 0。

3. 这个 12px 的限制用户是可以自行调整的，进入 chrome://settings/fonts 设置，滚动到最下方你就可以调整 12px 为其他值。

如果我一定要设置小于 12px 的字体怎么办？

1. Chrome 29 版本之前，你可以使用 `-webkit-text-size-adjust: none;` 来解除这个限制。29 版本后，就不能这样做了。

2. 你可以先设置 12px，然后使用 `transform: scale(0.833333)` 讲元素缩小，效果跟 10px 很接近。不过要注意的是，`transform: scale` 出了缩小 font-size，也会缩小其他一些属性，需要多测试。

4.3.15、button.disabled 和 button.getAttribute('disabled') 有什么区别？

「`node.getAttribute('someAttribute')`」获取的是「attribute」，而「`node.someAttribute`」获取的是元素的「property」，二者并不相同

「`node.property`」的方式不能获取自定义属性，「`node.getAttribute()`」的方式可以获取自定义属性

「`node.getAttribute()`」获取自定义属性忽略属性的大小写

「`node.getAttribute()`」获取自定义属性得到的值的类型总是字符串

4.3.16、var undefined = 1 这样赋值有效果吗？在什么情况下有？Why？

情景 1：

```
var undefined = 1;
alert(undefined); // chrome: undefined, ie8: 1
alert(window.undefined);
```

在 chrome 下运行得到的结果还是「undefined」，但在 IE8 以下赋值是生效的。在 ES3 中(Firefox4 之前),window.undefined 就是一个普通的属性,你完全可以把它值改变成为任意的真值,但在 ES5 中((Firefox4 之后),window.undefined 成了一个不可写,不可配置的数据属性,它的值永远是 undefined。

情景 2：

```
var obj = {};
obj.undefined = 'jirengu ruoyu';
console.log(obj.undefined) // jirengu ruoyu
```

在标准浏览器下作为全局作用域下 window 的一个属性，undefined 不可修改；但对于一个普通对象，undefined 可作为属性且可以修改

情景 3：

```
function fn(){
  var undefined = 100;
  alert(undefined); //chrome: 100, ie8: 100
}
```


fn();

不管是标准浏览器，还是老的 IE 浏览器，在函数内部 `undefined` 可作为局部变量重新赋值

情景 4:

```
undefined = 100;
(function(global){
    alert(undefined); //chrome: undefined, ie8: 100
})(window)
```

在标准浏览器下输出的结果是 `undefined`。而在 ie8 下为 100（因为可被赋值）

4.3.17、为什么 `0.1 + 0.2` 结果为 `0.30000000000000004`?

如果你打开浏览器的控制台，输入以下代码并运行

```
> 0.1 + 0.2
```

```
0.30000000000000004
```

奇怪，为什么不是 0.3。

计算机将所有数据以二进制的形式存储

计算机用有限的大小来存储数据（因为现实生活中不存在无限大的内存或硬盘）

十进制的 0.1 转为二进制，得到一个无限循环小数：`0.00011...`。也就是说，二进制无法「用有限的位数」来表示 0.1。对于 0.2 也是一样的，不赘述。二进制能「用有限的位数」表示的有：0.5、0.25、0.125 等。

但是计算机只能用有限的位数来存一个数，所以最终，计算机存的数是一个近似于 0.1 的小数。确切的二进制值是 `0-01111111011-10011001100110011001100110011001100110011001100110011010`，对应的十进制值是 `0.10000000000000000555...`。这是计算机能表示的最接近 0.1 的数字（存储一个数字的空间越大则越精确）。所以当我们计算 `0.1 + 0.2` 时，实际上算的是两个近似值相加，得到的值当然也是近似等于 0.3

4.3.18、`document.write` 作用

有人说页面加载完毕前调用 `document.write` 就不清空，调用完再调用 `document.write` 就会清空，是这样吗

当你打开一个页面，浏览器会

- 1.调用 `document.open()` 打开文档
- 2.`document.write(...)` 将下载到的网页内容写入文档
- 3.所有内容写完了，就调用 `document.close()`
- 4.触发 dom ready 事件（`DOMContentLoaded`）

所以如果你在第 3 步之前 `document.write(1)` 那么你就直接追加内容到当前位置，如果你在第 3 步之后 `document.write()`，那么由于 `document` 已经 `close` 了，所以必须重新 `document.open()` 来打开文档，这一打开，内容就被清空了。

不信你可以这样验证一下：

- 1.使用浏览器打开 html 文件 等页面加载完
- 2.在控制台运行 `document.write(1)`，会看到页面清空，只有一个 1
- 3.再次运行 `document.write(1)`，会发现页面没有清空，1 变成了 11，因为追加了一个 1
- 4.运行 `document.close()`，这是文档就关闭了。
- 5.再次运行 `document.write(1)`，你会发现文档又清空了，变成了 1。

4.4. 其他问题

4.4.1、移动端（Android IOS）怎么做好用户体验？

清晰的视觉纵线、信息的分组、极致的减法、
利用选择代替输入、标签及文字的排布方式、
依靠明文确认密码、合理的键盘利用

4.4.2、你在现在的团队处于什么样的角色，起到了什么明显的作用？

4.4.3、你认为怎样才是全端工程师（Full Stack developer）？

4.4.4、介绍一个你最得意的作品吧？

4.4.5、你的优点是什么？缺点是什么？

4.4.6、如何管理前端团队？

4.4.7、最近在学什么？能谈谈你未来 3，5 年给自己的规划吗？

面试注意点：

面试题目： 根据你的等级和职位变化，入门级到专家级：范围个、深度个、方向个。

题目类型： 技术视野、项目细节、理论知识题，算法题，开放性题，案例题。

进行追问： 可以确保问到你开始不懂或面试官开始不懂为止，这样可以大大延展题目的区分度和深度，知道你的实际能力。因为这种关联知识是长时期的学习，绝对不是临时记得住的。

回答问题再棒面试官（可能是你的直接领导面试），会考虑我要不要这个人做我的同事？所以态度很重要。（感觉更像是相亲）

资深的工程师能把 absolute 和 relative 弄混，这样的人不要也罢，因为团队需要的你这个人具有可以依靠的才能（靠谱）。

无限互联

无限互联官方 微信公众号



无限互联全栈研发中心 微信公众号



定价： 39.00 元