前端面试题目搜集

博客选编

目　录

一、理论知识

> 原文出处：http://www.cnblogs.com/strick/p/4968200.html
> 作者：咖啡机(K.F.J)

1.1、讲讲输入完网址按下回车，到看到网页这个过程中发生了什么

a. 域名解析

b. 发起TCP的3次握手

c. 建立TCP连接后发起http请求

d. 服务器端响应http请求，浏览器得到html代码

e. 浏览器解析html代码，并请求html代码中的资源

f. 浏览器对页面进行渲染呈现给用户

参考《一次完整的HTTP事务是怎样一个过程》

1.2、谈谈你对前端性能优化的理解

a. 请求数量：合并脚本和样式表，CSS Sprites，拆分初始化负载，划分主域

b. 请求带宽：开启GZip，精简JavaScript，移除重复脚本，图像优化，将icon做成字体

c. 缓存利用：使用CDN，使用外部JavaScript和CSS，添加Expires头，减少DNS查找，配置ETag，使AjaX可缓存

d. 页面结构：将样式表放在顶部，将脚本放在底部，尽早刷新文档的输出

e. 代码校验：避免CSS表达式，避免重定向

参考《前端工程与性能优化》

*1.3、前端 MV框架的意义**

早期前端都是比较简单，基本以页面为工作单元，内容以浏览型为主，也偶尔有简单的表单操作，基本不太需要框架。

随着 AJAX 的出现，Web2.0的兴起，人们可以在页面上可以做比较复杂的事情了，然后前端框架才真正出现了。

如果是页面型产品，多数确实不太需要它，因为页面中的 JavaScript代码，处理交互的绝对远远超过处理模型的，但是如果是应用软件类产品，这就太需要了。

长期做某个行业软件的公司，一般都会沉淀下来一些业务组件，主要体现在数据模型、业务规则和业务流程，这些组件基本都存在于后端，在前端很少有相应的组织。

从协作关系上讲，很多前端开发团队每个成员的职责不是很清晰，有了前端的 MV框架，这个状况会大有改观。

之所以感受不到 MV*框架的重要性，是因为Model部分代码较少，View的相对多一些。如果主要在操作View和Controller，那当然 jQuery 这类库比较好用了。

参考《前端 MV*框架的意义》

1.4、请简述盒模型

IE6盒子模型与W3C盒子模型。

文档中的每个元素被描绘为矩形盒子。盒子有四个边界：外边距边界margin, 边框边界border, 内边距边界padding与内容边界content。

CSS3中有个box-sizing属性可以控制盒子的计算方式，

content-box：padding和border不被包含在定义的width和height之内。对象的实际宽度等于设置的width值和border、padding之和。（W3C盒子模型）

border-box：padding和border被包含在定义的width和height之内。对象的实际宽度就等于设置的width值。（IE6盒子模型）

参考《盒模型》

1.5、请你谈谈Cookie的弊端

a. 每个特定的域名下最多生成的cookie个数有限制

b. IE和Opera 会清理近期最少使用的cookie，Firefox会随机清理cookie

c. cookie的最大大约为4096字节，为了兼容性，一般不能超过4095字节

d. 安全性问题。如果cookie被人拦截了，那人就可以取得所有的session信息。

1.6、浏览器本地存储

在HTML5中提供了sessionStorage和localStorage。

sessionStorage用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能

访问并且当会话结束后数据也随之销毁，是会话级别的存储。

localStorage用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

1.7、web storage和cookie的区别

a. Cookie的大小是受限的

b. 每次你请求一个新的页面的时候Cookie都会被发送过去，这样无形中浪费了带宽

c. cookie还需要指定作用域，不可以跨域调用

d. Web Storage拥有setItem,getItem等方法，cookie需要前端开发者自己封装setCookie，getCookie

e. Cookie的作用是与服务器进行交互，作为HTTP规范的一部分而存在 ，而Web Storage仅仅是为了在本地“存储”数据而生

f. IE7、IE6中的UserData通过简单的代码封装可以统一到所有的浏览器都支持web storage

1.8、对BFC规范的理解

BFC全称是Block Formatting Context，即块格式化上下文。它是CSS2.1规范定义的，关于CSS渲染定位的一个概念。

BFC是页面CSS 视觉渲染的一部分，用于决定块盒子的布局及浮动相互影响范围的一个区域。

BFC的一个最重要的效果是，让处于BFC内部的元素与外部的元素相互隔离，使内外元素的定位不会相互影响。

利用BFC可以闭合浮动，防止与浮动元素重叠。

参考《Learning BFC》

1.9、线程与进程的区别

a. 一个程序至少有一个进程，一个进程至少有一个线程

b. 线程的划分尺度小于进程，使得多线程程序的并发性高

c. 进程在执行过程中拥有独立的内存单元，而多个线程共享内存，从而极大地提高了程序的运行效率

d. 每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口。但是线程不能够独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制

e. 多线程的意义在于一个应用程序中，有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用，来实现进程的调度和管理以及资源分配

1.10、请说出三种减少页面加载时间的方法

a. 尽量减少页面中重复的HTTP请求数量

b. 服务器开启gzip压缩

c. css样式的定义放置在文件头部

d. Javascript脚本放在文件末尾

e. 压缩合并Javascript、CSS代码

f. 使用多域名负载网页内的多个文件、图片

参考《减低页面加载时间的方法》

1.11、你都使用哪些工具来测试代码的性能？

JSPerf, Dromaeo

1.12、你遇到过比较难的技术问题是？你是如何解决的？

1.13、常使用的库有哪些？常用的前端开发工具？开发过什么应用或组件？

1.14、列举IE与其他浏览器不一样的特性？

a. IE的排版引擎是Trident（又称为MSHTML）

b. Trident内核曾经几乎与W3C标准脱节（2005年）

c. Trident内核的大量 Bug等安全性问题没有得到及时解决

d. JS方面，有很多独立的方法，例如绑定事件的attachEvent、创建事件的createEventObject等

e. CSS方面，也有自己独有的处理方式，例如设置透明，低版本IE中使用滤镜的方式

参考《Trident（排版引擎）》

1.15、什么叫优雅降级和渐进增强？

渐进增强 progressive enhancement：

针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

优雅降级 graceful degradation：

一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

区别：

a. 优雅降级是从复杂的现状开始，并试图减少用户体验的供给

b. 渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要

c. 降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带

参考《优雅降级和渐进增强的区别》

1.16、WEB应用从服务器主动推送Data到客户端有那些方式？

a. html5 websoket

b. WebSocket 通过 Flash

c. XHR长时间连接

d. XHR Multipart Streaming

e. 不可见的Iframe

f. 标签的长时间连接(可跨域)

1.17、对前端界面工程师这个职位是怎么样理解的？

a. 前端是最贴近用户的程序员，前端的能力就是能让产品从 90分进化到 100 分，甚至更好

b. 参与项目，快速高质量完成实现效果图，精确到1px；

c. 与团队成员，UI设计，产品经理的沟通；

d. 做好的页面结构，页面重构和用户体验；

e. 处理hack，兼容、写出优美的代码格式；

f. 针对服务器的优化、拥抱最新前端技术。

1.18、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？

1.19、你的优点是什么？缺点是什么？

1.20、如何管理前端团队?

1.21、最近在学什么？能谈谈你未来3，5年给自己的规划吗？

1.22、平时如何管理你的项目？

a. 先期团队必须确定好全局样式（globe.css），编码模式(utf-8) 等；

b. 编写习惯必须一致（例如都是采用继承式的写法，单样式都写成一行）；

c. 标注样式编写人，各模块都及时标注（标注关键样式调用的地方）；

d. 页面进行标注（例如 页面 模块 开始和结束）；

e. CSS跟HTML 分文件夹并行存放，命名都得统一（例如style.css）；

f. JS 分文件夹存放 命名以该JS功能为准的英文翻译。

g. 图片采用整合的 images.png png8 格式文件使用 尽量整合在一起使用方便将来的管理

1.23、说说最近最流行的一些东西吧？常去哪些网站？

CSDN、SegmentFault、php.net、MDN、css参考手册、iconfont、

underscore、github、Bootstrap、W3Shool、W3Cplus、caniuse

1.24、Flash、Ajax各自的优缺点，在使用中如何取舍？

Flash：
a. Flash适合处理多媒体、矢量图形、访问机器

b. 对CSS、处理文本上不足，不容易被搜索

Ajax：

a. Ajax对CSS、文本支持很好，支持搜索

b. 多媒体、矢量图形、机器访问不足

共同点：

a. 与服务器的无刷新传递消息

b. 可以检测用户离线和在线状态

c. 操作DOM

1.25、请解释一下 JavaScript 的同源策略

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。

指一段脚本只能读取来自同一来源的窗口和文档的属性。

1.26、AMD和CMD 规范的区别？

AMD 提前执行依赖 - 尽早执行，requireJS 是它的实现

CMD 按需执行依赖 - 懒执行，seaJS 是它的实现

参考《SeaJS与RequireJS最大的区别》、《与 RequireJS 的异同》

1.27、网站重构的理解

重构：在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。

a. 使网站前端兼容于现代浏览器(针对于不合规范的CSS、如对IE6有效的)

b. 对于移动平台的优化，针对于SEO进行优化

c. 减少代码间的耦合，让代码保持弹性

d. 压缩或合并JS、CSS、image等前端资源

1.28、浏览器的内核分别是什么?

IE浏览器的内核Trident、Mozilla的Gecko、Chrome的Blink（WebKit的分支）、Opera内核原为Presto，现为Blink；

1.29、请介绍下cache-control

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）。

1.30、前端页面有哪三层构成，分别是什么？作用是什么？

a. 结构层：由 HTML 或 XHTML 之类的标记语言负责创建，仅负责语义的表达。解决了页面“内容是什么”的问题。

b. 表示层：由CSS负责创建，解决了页面“如何显示内容”的问题。

c. 行为层：由脚本负责。解决了页面上“内容应该如何对事件作出反应”的问题。

1.31、知道的网页制作会用到的图片格式有哪些？

png-8，png-24，jpeg，gif，svg

Webp：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。

Apng：全称是“Animated Portable Network Graphics”，是PNG的位图动画扩展，可以实现png格式的动态图片效果。04年诞生，但一直得不到各大浏览器厂商的支持，直到日前得到 iOS safari 8的支持，有望代替GIF成为下一代动态图标准。

1.32、一次js请求一般情况下有哪些地方会有缓存处理？

a. 浏览器端存储

b. 浏览器端文件缓存

c. HTTP缓存304

d. 服务器端文件类型缓存

e. 表现层&DOM缓存

参考《一次HTTP请求中有哪些地方可以缓存》

1.33、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。

a. 图片懒加载，滚动到相应位置才加载图片。

b. 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。

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

d. 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。

1.34、谈谈以前端角度出发做好SEO需要考虑什么？

a. 了解搜索引擎如何抓取网页和如何索引网页

b. meta标签优化

c. 关键词分析

d. 付费给搜索引擎

e. 链接交换和链接广泛度（Link Popularity）

f. 合理的标签使用

二、HTML

2.1、`<image>`标签上title属性与alt属性的区别是什么？

alt属性是为了给那些不能看到你文档中图像的浏览者提供文字说明的。且长度必须少于100个英文字符或者用户必须保证替换文字尽可能的短。

这包括那些使用本来就不支持图像显示或者图像显示被关闭的浏览器的用户，视觉障碍的用户和使用屏幕阅读器的用户等。

title属性为设置该属性的元素提供建议性的信息。使用title属性提供非本质的额外信息。参考《alt和title属性的区别及应用》

2.2、分别写出以下几个HTML标签：文字加粗、下标、居中、字体

加粗：`<b>`、`<strong>`

下标：`<sub>`

居中：`<center>`

字体：`<font>`、`<basefont>`、参考《HTML标签列表》

2.3、请写出至少5个html5新增的标签，并说明其语义和应用场景

section：定义文档中的一个章节

nav：定义只包含导航链接的章节

header：定义页面或章节的头部。它经常包含 logo、页面标题和导航性的目录。

footer：定义页面或章节的尾部。它经常包含版权信息、法律信息链接和反馈建议用的地址。

aside：定义和页面内容关联度较低的内容——如果被删除，剩下的内容仍然很合理。

参考《HTML5 标签列表》

2.4、请说说你对标签语义化的理解？

a. 去掉或者丢失样式的时候能够让页面呈现出清晰的结构

b. 有利于SEO：和搜索引擎建立良好沟通，有助于爬虫抓取更多的有效信息：爬虫依赖于标签来确定上下文和各个关键字的权重；

c. 方便其他设备解析（如屏幕阅读器、盲人阅读器、移动设备）以意义的方式来渲染网页；

d. 便于团队开发和维护，语义化更具可读性，遵循W3C标准的团队都遵循这个标准，可以减少差异化。

2.5、Doctype作用? 严格模式与混杂模式如何区分？它们有何意义?

声明位于文档中的最前面，处于 标签之前。告知浏览器以何种模式来渲染文档。

严格模式的排版和 JS 运作模式是，以该浏览器支持的最高标准运行。

在混杂模式中，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。

DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。

2.6、你知道多少种Doctype文档类型？

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。

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

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，

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。

2.7、HTML与XHTML——二者有什么区别

a. XHTML 元素必须被正确地嵌套。

b. XHTML 元素必须被关闭。

c. 标签名必须用小写字母。

d. XHTML 文档必须拥有根元素。

参考《XHTML 与 HTML 之间的差异》

2.8、html5有哪些新特性、移除了那些元素？

a. 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。

b. 拖拽释放(Drag and drop) API

c. 语义化更好的内容标签（header,nav,footer,aside,article,section）

d. 音频、视频API(audio,video)

e. 画布(Canvas) API

f. 地理(Geolocation) API

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

h. sessionStorage 的数据在页面会话结束时会被清除

i. 表单控件，calendar、date、time、email、url、search

j. 新的技术webworker, websocket等

移除的元素：

a. 纯表现的元素：basefont，big，center, s，strike，tt，u；

b. 对可用性产生负面影响的元素：frame，frameset，noframes；

2.9、iframe的优缺点？

优点：

a. 解决加载缓慢的第三方内容如图标和广告等的加载问题

b. iframe无刷新文件上传

c. iframe跨域通信

缺点：

a. iframe会阻塞主页面的Onload事件

b. 无法被一些搜索引擎索引到

c. 页面会增加服务器的http请求

d. 会产生很多页面，不容易管理。

参考《iframe的一些记录》

2.10、Quirks模式是什么？它和Standards模式有什么区别？

在写程序时我们也会经常遇到这样的问题，如何保证原来的接口不变，又提供更强大的功能，尤其是新功能不兼容旧功能时。IE6以前的页面大家都不会去写DTD，所以IE6就假定 如果写了DTD，就意味着这个页面将采用对CSS支持更好的布局，而如果没有，则采用兼容之前的布局方式。这就是Quirks模式（怪癖模式，诡异模式，怪异模式）。

区别：总体会有布局、样式解析和脚本执行三个方面的区别。

a. 盒模型：在W3C标准中，如果设置一个元素的宽度和高度，指的是元素内容的宽度和高度，而在Quirks 模式下，IE的宽度和高度还包含了padding和border。

b. 设置行内元素的高宽：在Standards模式下，给等行内元素设置wdith和height都不会生效，而在quirks模式下，则会生效。

c. 设置百分比的高度：在standards模式下，一个元素的高度是由其包含的内容来决定的，如果父元素没有设置百分比的高度，子元素设置一个百分比的高度是无效的用

d. 设置水平居中：使用margin:0 auto在standards模式下可以使元素水平居中，但在quirks模式下却会失效。

2.11、请阐述table的缺点

a. 太深的嵌套，比如table>tr>td>h3，会导致搜索引擎读取困难，而且，最直接的损失就是大大增加了冗余代码量。

b. 灵活性差，比如要将tr设置border等属性，是不行的，得通过td

c. 代码臃肿，当在table中套用table的时候，阅读代码会显得异常混乱

d. 混乱的colspan与rowspan，用来布局时，频繁使用他们会造成整个文档顺序混乱。

e. 不够语义

参考《为什么说table表格布局不好？》

2.12、简述一下src与href的区别

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）或当前文档（链接）之间的链接

三、CSS

3.1、谈谈你对CSS布局的理解

3.2、请列举几种可以清除浮动的方法（至少两种）

浮动会漂浮于普通流之上，像浮云一样，但是只能左右浮动。正是这种特性，导致框内部由于不存在其他普通流元素了，表现出高度为0（高度塌陷）。

a. 添加额外标签，例如`<div style="clear:both"></div>`

b. 使用br标签和其自身的html属性，例如`<br clear="all" />`

c. 父元素设置 overflow：hidden；在IE6中还需要触发hasLayout，例如zoom：1；

d. 父元素设置 overflow：auto 属性；同样IE6需要触发hasLayout

e. 父元素也设置浮动

f. 父元素设置display:table

g. 使用:after 伪元素；由于IE6-7不支持:after，使用 zoom:1触发 hasLayout。

在CSS2.1里面有一个很重要的概念，那就是 Block formatting contexts（块级格式化上下文），简称BFC。

创建了BFC的元素就是一个独立的盒子，里面的子元素不会在布局上影响外面的元素，同时BFC仍然属于文档中的普通流。

IE6-7的显示引擎使用的是一个称为布局（layout）的内部概念。

参考《那些年我们一起清除过的浮动》

3.3、请列举几种隐藏元素的方法

a. visibility: hidden；这个属性只是简单的隐藏某个元素，但是元素占用的空间任然存在。

b. opacity: 0；一个CSS3属性，设置0可以使一个元素完全透明，制作出和visibility一样的效果。与visibility相比，它可以被transition和animate

c. position: absolute；使元素脱离文档流，处于普通文档之上，给它设置一个很大的left负值定位，使元素定位在可见区域之外。

d. display: none；元素会变得不可见，并且不会再占用文档的空间。

e. transform: scale(0)；将一个元素设置为无限小，这个元素将不可见。这个元素原来所在的位置将被保留。

f. HTML5 hidden attribute；hidden属性的效果和display:none;相同，这个属性用于记录一个元素的状态

g. height: 0; overflow: hidden；将元素在垂直方向上收缩为0,使元素消失。只要元素没有可见的边框，该技术就可以正常工作。

h. filter: blur(0)；将一个元素的模糊度设置为0，从而使这个元素“消失”在页面中。

参考《使用CSS隐藏HTML元素的4种常用方法》《通过HTML和CSS隐藏和显示元素的4种方法》

3.4、如何让一段文本中的所有英文单词的首字母大写

text-transform：

none| capitalize(将每个单词的第一个字母转换成大写) | uppercase(将每个单词转换成大写) | lowercase(将每个单词转换成小写)

3.5、请简述CSS样式表继承

CSS样式表继承指的是，特定的CSS属性向下传递到子孙元素。会被继承下去的属性如下：参考《CSS样式表继承详解》

文本相关：font-family，font-size， font-style，font-variant，font-weight， font，letter-spacing，line-height，color

列表相关：list-style-image，list-style-position，list-style-type， list-style

3.6、请简述CSS的选择器

元素选择器：* 、E、 E#id、 E.class

关系选择器：E、F、E>F、E+F、E~F

属性选择器：E[att]、E[att="val"]、E[att~="val"]、E[att^="val"]、E[att$="val"]、E[att*="val"]、E[att|="val"]

伪类选择器：E:link、E:visited、E:hover、E:active、E:focus、E:lang(fr)、E:not(s)、E:root、E:first-child、E:last-chil等

伪对象选择器：E:first-letter/E::first-letter、E:first-line/E::first-line、E:before/E::before、E:after/E::after、E::selection

参考《选择符列表》

3.7、CSS伪类与CSS伪对象的区别

CSS 引入伪类和伪元素的概念是为了描述一些现有CSS无法描述的东西

根本区别在于：它们是否创造了新的元素（抽象）

伪类：一开始用来表示一些元素的动态状态，随后CSS2标准扩展了其概念范围，使其成为了所有逻辑上存在但在文档树中却无须标识的“幽灵”分类

伪对象：代表了某个元素的子元素，这个子元素虽然在逻辑上存在，但却并不实际存在于文档树中

参考《CSS伪类与CSS伪元素的区别及由来》

3.8、请简述CSS的权重规则

一个行内样式+1000，一个id+100，一个属性选择器/class类/伪类选择器+10，一个元素名/伪对象选择器+1。

关系选择器将拆分为两个选择器再计算。参考《CSS权重》

3.9、请写出多种等高布局

a. 假等高列：使用背景图片，在列的父元素上使用这个背景图进行Y轴的铺放，从而实现一种等高列的假像

b. 给容器div使用单独的背景色（固定布局）（流体布局）：用元素中的最大高度撑大其他的容器高度

c. 创建带边框的两列等高布局：用border-left来做，只能使用两列。

d. 使用正padding和负margin对冲实现多列布局方法：在所有列中使用正的上、下padding和负的上、下margin，并在所有列外面加上一个容器，设置overflow:hiden把溢出背景切掉

e. 使用边框和定位模拟列等高：但不能使用在多列

f. 模仿表格布局等高列效果：兼容性不好，在ie6-7无法正常运行

3.10、在CSS样式中常使用px、em，各有什么优劣，在表现上有什么区别？

px是相对长度单位，相对于显示器屏幕分辨率而言的。

em是相对长度单位，相对于当前对象内文本的字体尺寸。

px定义的字体，无法用浏览器字体放大功能。

em的值并不是固定的，会继承父级元素的字体大小，1 ÷ 父元素的font-size × 需要转换的像素值 = em值。

3.11、CSS中 link 和@import 的区别是什么？

a. link属于HTML标签，而@import是CSS提供的，且只能加载 CSS

b. 页面被加载时，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

c. import只在IE5以上才能识别，而link是HTML标签，无兼容问题

d. link方式的样式的权重 高于@import的权重

e. 当使用 Javascript 控制 DOM 去改变样式的时候，只能使用 link 方式，因为 @import 眼里只有 CSS，不是 DOM 可以控制

3.12、position的absolute与fixed共同点与不同点

相同：

a. 改变行内元素的呈现方式，display被置为block

b. 让元素脱离普通流，不占据空间

c. 默认会覆盖到非定位元素上

区别：

absolute的”根元素“是可以设置的，而fixed的”根元素“固定为浏览器窗口。

当你滚动网页，fixed元素与浏览器窗口之间的距离是不变的。

3.13、position的值， relative和absolute分别是相对于谁进行定位的？

absolute：生成绝对定位的元素，相对于 static 定位以外的第一个祖先元素进行定位

fixed：生成绝对定位的元素，相对于浏览器窗口进行定位。（IE6不支持）

relative：生成相对定位的元素，相对于其在普通流中的位置进行定位

static：默认值。没有定位，元素出现在正常的流中

3.14、CSS3有哪些新特性？

CSS3实现圆角（border-radius），阴影（box-shadow），对文字加特效（text-shadow），线性渐变（gradient），变形（transform）

增加了更多的CSS选择器 多背景 rgba，在CSS3中唯一引入的伪元素是::selection，媒体查询，多栏布局

参考《CSS3中的动画效果记录》、《CSS3中border-radius、box-shadow与gradient那点事儿》

3.15、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。

3.16、解释下 CSS sprites原理，优缺点

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可以用数字精确的定位出背景图片的位置。

优点：

a. 减少网页的http请求

b. 减少图片的字节

c. 解决了网页设计师在图片命名上的困扰，只需对一张集合的图片上命名就可以了，不需要对每一个小元素进行命名

d. 更换风格方便，只需要在一张或少张图片上修改图片的颜色或样式，整个网页的风格就可以改变。

缺点：

a. 在宽屏，高分辨率的屏幕下的自适应页面，你的图片如果不够宽，很容易出现背景断裂

b. CSS Sprites在开发的时候，要通过photoshop或其他工具测量计算每一个背景单元的精确位置

c. 在维护的时候比较麻烦，如果页面背景有少许改动，一般就要改这张合并的图片

3.17、解释下浮动和它的工作原理？

a. 浮动元素脱离文档流，不占据空间（引起“高度塌陷”现象）

b. 浮动元素碰到包含它的边框或者浮动元素的边框停留。

3.18、浮动元素引起的问题

a. 父元素的高度无法被撑开，影响与父元素同级的元素

b. 与浮动元素同级的非浮动元素会跟随其后

c. 若非第一个元素浮动，则该元素之前的元素也需要浮动，否则会影响页面显示的结构

3.19、什么是 FOUC（无样式内容闪烁）？你如何来避免 FOUC？

如果使用import方法对CSS进行导入，会导致某些页面在Windows下的IE出现一些奇怪的现象：

以无样式显示页面内容的瞬间闪烁，这种现象称之为文档样式短暂失效(Flash of Unstyled Content)，简称为FOUC。

原理：当样式表晚于结构性html加载，当加载到此样式表时，页面将停止之前的渲染。此样式表被下载和解析后，将重新渲染页面，也就出现了短暂的花屏现象。

解决方法：使用LINK标签将样式表放在文档HEAD中。

3.20、line-height三种赋值方式有何区别？（带单位、纯数字、百分比）

带单位：px不用计算，em则会使元素以其父元素font-size值为参考来计算自己的行高

纯数字：把比例传递给后代，例如父级行高为1.5，子元素字体为18px，则子元素行高为1.5*18=27px

百分比：将计算后的值传递给后代

参考《line-height的理解》、《浅析line-height和vertical》，查看在线源码。

3.21、:link、:visited、:hover、:active的执行顺序是怎么样的？

L-V-H-A，l(link)ov(visited)e h(hover)a(active)te，即用喜欢和讨厌两个词来概括

3.22、经常遇到的浏览器兼容性有哪些？如何解决？

a. 浏览器默认的margin和padding不同

b. IE6双边距bug

c. 在ie6，ie7中元素高度超出自己设置高度。原因是IE8以前的浏览器中会给元素设置默认的行高的高度导致的

d. min-height在IE6下不起作用

e. 透明性IE用filter:Alpha(Opacity=60)，而其他主流浏览器用 opacity:0.6

f. input边框问题，去掉input边框一般用border:none;就可以，但由于IE6在解析input样式时的BUG(优先级问题)，在IE6下无效

3.23、有哪项方式可以对一个DOM设置它的CSS样式？

a. 外部样式表：通过 `<link>` 标签引入一个外部css文件

b. 内部样式表：将css代码放在 `<style>` 标签内部

c. 内联样式：将css样式直接定义在 HTML 元素内部

3.24、什么是外边距重叠？重叠的结果是什么？

外边距重叠就是margin-collapse。

在CSS当中，相邻的两个盒子（可能是兄弟关系也可能是祖先关系）的外边距可以结合成一个单独的外边距。这种合并外边距的方式被称为折叠，并且因而所结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则：

a. 两个相邻的外边距都是正数时，折叠结果是它们两者之间较大的值。

b. 两个相邻的外边距都是负数时，折叠结果是两者绝对值的较大值。

c. 两个外边距一正一负时，折叠结果是两者的相加的和。

3.25、rgba()和opacity的透明效果有什么不同？

a. opacity作用于元素，以及元素内的所有内容的透明度，rgba()只作用于元素的颜色或其背景色。

b. 设置rgba透明的元素的子元素不会继承透明效果！

3.26、css属性content有什么作用？有什么应用？

css的content属性专门应用在 before/after 伪元素上，用于来插入生成内容。

可以配合自定义字体显示特殊符号。

四、JavaScript

4.1、请解释一下什么是闭包

闭包是一种特殊的对象。它由两部分构成：函数，以及创建该函数的环境。

可以把闭包简单理解成 "定义在一个函数内部的函数"，闭包就是将函数内部和函数外部连接起来的一座桥梁。闭包有如下特性：

a. JavaScript允许你使用在当前函数以外定义的变量

b. 即使外部函数已经返回，当前函数仍然可以引用在外部函数所定义的变量

c. 闭包可以更新外部变量的值

d. 用闭包模拟私有方法

由于闭包会使得函数中的变量都被保存在内存中，内存消耗很大，所以不能滥用闭包，否则会造成网页的性能问题

4.2、call 和 apply 的区别是什么？

call 和 apply 就是为了改变函数体内部 this 的指向。

区别是从第二个参数起，call 需要把参数按顺序传递进去，而 apply 则是把参数放在数组里。

当参数明确时用call与apply都行, 当参数不明确时可用apply给合arguments

4.3、如何使用原生 Javascript 代码深度克隆一个对象（注意区分对象类型）

在网上找了个函数，用递归的方式做复制。传入的参数必须得是Array或Object。

并且用到了JSON.stringify和JSON.parse。查看在线代码。参考《JavaScript中的对象克隆》

4.4、 jQuery中 $('.class')和$('div.class') 哪个效率更高？

jQuery内部使用Sizzle引擎，处理各种选择器。Sizzle引擎的选择顺序是从右到左，所以这条语句是先选.class，

第二个会直接过滤出div标签，而第一个就不会过滤了，将所有相关标签都列出。参考《jQuery最佳实践》

4.5、实现输出document对象中所有成员的名称和类型

用一个for in方式循环document，然后在将内容console出来，

就是看到篇文章还会判断document.hasOwnProperty，然后再做打印，我测试了下这样的话打印不出来。

查看在线代码。参考《JavaScript要点归档：DOM》

4.6、获得一个DOM元素的绝对位置

offsetTop：返回当前元素相对于其 offsetParent 元素的顶部的距离

offsetLeft：返回当前元素相对于其 offsetParent 元素的左边的距离

getBoundingClientRect()：返回值是一个DOMRect对象，它包含了一组用于描述边框的只读属性——left、top、right和bottom，属性单位为像素

参考《JavaScript中尺寸、坐标》，查看在线代码。

4.7、如何利用JS生成一个table？

首先是用createElement创建一个table，再用setAttribute设置table的属性，

然后用for循环设置tr和td的内容，用appendChild拼接内容，设置td的时候还用到innerHTML和style.padding。

查看在线代码。参考《JavaScript要点归档：DOM表格》《JavaScript要点归档：DOM》

4.8、实现预加载一张图片，加载完成后显示在网页中并设定其高度为50px，宽度为50px

先new Image()获取一个图片对象，然后在图片对象的onload中设置宽度和高度。查看在线代码。

4.9、假设有一个4行tr的table，将table里面tr顺序颠倒

先是通过table.tBodies[0].rows获取到当前tbody中的行，接下来是两种方法处理。获取到的行没有reverse这个方法。

第一种是将这些行push到另外一个数组中

第二种是用Array.prototype.slice.call()将那些行变成数组，

接着用reverse倒叙，table再appendChild。查看在线代码。

这里我有个疑问，就是在appendChild的时候，并不是在最后把列加上，而是做了替换操作？

4.10、模拟一个HashTable类，一个类上注册四个方法：包含有add、remove、contains、length方法

先是在构造函数中定义一个数组，然后用push模拟add，splice模拟remove。

四个方法都放在了prototype上面。查看在线代码。

4.11、Ajax读取一个XML文档并进行解析的实例

a. 初始化一个HTTP请求，IE以ActiveX对象引入。 后来标准浏览器提供了XMLHttpRequest类，它支持ActiveX对象所提供的方法和属性

b. 发送请求，可以调用HTTP请求类的open()和send()方法

c. 处理服务器的响应，通过http_request.onreadystatechange = nameOfTheFunction。来指定函数

参考《AJAX》《开始AJAX》，查看在线代码。

4.12、JS如何实现面向对象和继承机制？

创建对象方法：

a. 利用json创建对象

b. 使用JavaScript中的Object类型

c. 通过创建函数来生成对象

继承机制：

a. 构造函数绑定，使用call或apply方法，将父对象的构造函数绑定在子对象上

b. prototype模式，继承new函数的模式

c. 直接继承函数的prototype属性，对b的一种改进

d. 利用空对象作为中介

e. 在ECMAScript5中定义了一个新方法Object.create()，用于创建一个新方法

f. 拷贝继承，把父对象的所有属性和方法，拷贝进子对象，实现继承。参考《JavaScript中的对象克隆》

参考《Javascript继承机制的设计思想》《构造函数的继承》，查看在线代码。

4.13、JS模块的封装方法，比如怎样实现私有变量，不能直接赋值，只能通过公有方法

a. 通过json生成对象的原始模式，多写几个就会非常麻烦，也不能反映出它们是同一个原型对象的实例

b. 原始模式的改进，可以写一个函数，解决代码重复的问题。同样不能反映出它们是同一个原型对象的实例

c. 构造函数模式，就是一个普通函数，不过内部使用了this变量，但是存在一个浪费内存的问题。

d. Prototype模式，每一个构造函数都有一个prototype属性，指向另一个对象。这个对象的所有属性和

方法，都会被构造函数的实例继承，可以把那些不变的属性和方法，直接定义在prototype对象上。Prototype模式的验证方法：isPrototypeOf()、hasOwnProperty()和in运算符。

参考《封装》，查看在线代码。

4.14、对this指针的理解，可以列举几种使用情况？

this指的是：调用函数的那个对象。

a. 纯粹的函数调用，属于全局性调用，因此this就代表全局对象Global。

b. 作为对象方法的调用，这时this就指这个上级对象。

c. 作为构造函数调用，就是通过这个函数new一个新对象（object）。这时，this就指这个新对象。

d. apply与call的调用，它们的作用是改变函数的调用对象，它的第一个参数就表示改变后的调用这个函数的对象。

参考《Javascript的this用法》，查看在线代码。

4.15、在JavaScript中，常用的绑定事件的方法有哪些？

a. 在DOM元素中直接绑定，DOM元素，可以理解为HTML标签，`onXXX="JavaScript Code"`，查看事件列表。

b. 在JavaScript代码中绑定，`elementObject.onXXX=function(){}`，通称为DOM0事件系统。

c. 绑定事件监听函数，标准浏览器使用 addEventListener() ，IE11以下版本attachEvent() 来绑定事件监听函数，通称为DOM2事件系统。

参考《JavaScript绑定事件的方法》

4.16、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

参考《事件》、《javascript的冒泡和捕获》，查看在线代码。

4.17、jQuery的特点

a. 一款轻量级的js库

b. 丰富快速的DOM选择器

c. 链式表达式

d. 事件、样式、动画等特效支持

e. Ajax操作封装，支持跨域

f. 跨浏览器兼容

g. 插件扩展开发

参考《JQuery特点、优缺点及其常用操作》

4.18、Ajax有哪些好处和弊端？

优点：

a. 无刷新更新数据

b. 异步与服务器通信

c. 前端和后端负载平衡

d. 基于标准被广泛支持

e. 界面与应用分离

缺点：

a. AJAX干掉了Back和History功能，即对浏览器机制的破坏

b. AJAX的安全问题

c. 对搜索引擎支持较弱

d. 违背URL和资源定位的初衷

参考《AJAX工作原理及其优缺点》

4.19、null和undefined的区别？

null：

a. null是一个表示"无"的对象，转为数值时为0

b. null表示"没有对象"，即该处不应该有值。

undefined：

a. undefined是一个表示"无"的原始值，转为数值时为NaN。

b. undefined表示"缺少值"，就是此处应该有一个值，但是还没有定义。

参考《undefined与null的区别》

4.20、new操作符具体干了什么呢?

a. 一个新对象被创建。它继承自函数原型

b. 构造函数被执行。执行的时候，相应的传参会被传入

c. 上下文(this)会被指定为这个新实例

d. 如果构造函数返回了一个“对象”，那么这个对象会取代整个new出来的结果

参考《new运算符》

4.21、js延迟加载的方式有哪些？

a. 将script节点放置在最后之前

b. 使用script标签的defer和async属性，defer属性为延迟加载，是在页面渲染完成之后再进行加载的，而async属性则是和文档并行加载

c. 通过监听onload事件，动态添加script节点

d. 通过ajax下载js脚本，动态添加script节点

参考《javascript延迟加载方式》

4.22、如何解决跨域问题?

a. JSONP（JSON with Padding），填充式JSON

b. iframe跨域

c. HTML5的window.postMessage方法跨域

d. 通过设置img的src属性，进行跨域请求

e. 跨域资源共享（CORS），服务器设置Access-Control-Allow-OriginHTTP响应头之后，浏览器将会允许跨域请求

4.23、documen.write和 innerHTML的区别

write：

a. 改变 HTML 输出流

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. 输入css的style标签能改变样式，例如
`document.write("*b{color:red;font-weight:bold;}*");`

innerHTML：

a. 改变 HTML 内容

b. 输入css的style标签不能改变样式

参考《JavaScript HTML DOM - 改变 HTML》

4.24、哪些操作会造成内存泄漏？

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

b. 在IE中，如果循环引用中的任何对象是 DOM 节点或者 ActiveX 对象，垃圾收集系统则不会处理。

c. 闭包可以维持函数内局部变量，使其得不到释放。

d. 在销毁对象的时候，要遍历属性中属性，依次删除，否则会泄漏。

参考《js内存泄漏的几种情况》、《JavaScript内存分析》

4.25、JavaScript中的变量声明提升？

函数声明和变量声明总是被JavaScript解释器隐式地提升到包含他们的作用域的最顶端。

函数表达式中只会提升名称，函数体只有在执行到赋值语句时才会被赋值。

```
function foo() {
    bar();
    var x = 1;
}
function foo() {//等同于
```

```
    var x;
    bar();
    x = 1;
}
function test() {
    foo(); // TypeError "foo is not a function"
    bar(); // "this will run!"
    var foo = function () { }// 函数表达式被赋值给变量'foo'
    function bar() { }// 名为'bar'的函数声明
}
```

4.26、如何判断当前脚本运行在浏览器还是node环境中？

通过判断Global对象是否为window，如果是window，当前脚本运行在浏览器中

4.27、什么是 "use strict"

ECMAscript 5添加了第二种运行模式："严格模式"（strict mode）

设立"严格模式"的目的，主要有以下几个：

a. 消除Javascript语法的一些不合理、不严谨之处，减少一些怪异行为;

b. 消除代码运行的一些不安全之处，保证代码运行的安全；

c. 提高编译器效率，增加运行速度；

d. 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

参考《Javascript 严格模式详解》

4.28、eval是做什么的？

eval()函数可计算某个字符串，并执行其中的的 JavaScript 代码。

eval()是一个顶级函数并且跟任何对象无关。

如果字符串表示了一个表达式，eval()会对表达式求值。如果参数表示了一个或多个JavaScript声明， 那么eval()会执行声明。

4.29、JavaScript原型，原型链？

原型：

a. 原型是一个对象，其他对象可以通过它实现属性继承。

b. 一个对象的真正原型是被对象内部的[[Prototype]]属性(property)所持有。浏览器支持非标准的访问

器proto。

c. 在javascript中，一个对象就是任何无序键值对的集合，如果它不是一个主数据类型(undefined，null，boolean，number，string)，那它就是一个对象。

原型链：

a. 因为每个对象和原型都有一个原型(注:原型也是一个对象)，对象的原型指向对象的父，而父的原型又指向父的父，我们把这种通过原型层层连接起来的关系撑为原型链。

b. 这条链的末端一般总是默认的对象原型。

```
a.__proto__ = b;
b.__proto__ = c;
c.__proto__ = {}; //default object
{}.__proto__.__proto__; //null
```

参考《理解JavaScript原型》

4.30、画出此对象的内存图

查看在线代码。

4.31、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

4.32、jQuery的源码看过吗？能不能简单说一下它的实现原理？

jQuery给我们带来了一个简洁方便的编码模型(1>创建jQuery对象;2>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

参考《jQuery工作原理解析以及源代码示例》

4.33、jQuery 中如何将数组转化为json字符串

在jQuery1.8.3中有个方法“parseJSON”，在这个方法中会做从string转换为json。

如果当前浏览器支持window.JSON，那就直接调用这个对象中的方法。

如果没有就使用(new Function("return " + data))();执行代码返回。

eval和new Function是有区别的。

4.34、请写出console.log中的内容

```
var msg = 'hello';//顶级作用域window下有个变量msg
function great(name, attr) {
 var name = 'david';
 var greating = msg + name + '!';
 var msg = '你好';
 for (var i = 0; i < 10; i++) {
     var next = msg + '你的id是' + i * 2 + i;
 }
 console.log(arguments[0]);
 console.log(arguments[1]);
 console.log(greating);
 console.log(next);
}
great('Tom')
```

查看在线代码。

a. arguments[0]被覆盖了

b. msg出现了声明提升，可以查看4.25的例子

c. next中出现了隐式的类型转换

4.35、请说明下下面代码的执行过程

```
var t=true;
window.setTimeout(function(){
     t=false;
},1000);
while(t){
  console.log(1);
}
alert('end');
```

查看在线代码。

a. JavaScript引擎是单线程运行的，浏览器无论在什么时候都只且只有一个线程在运行JavaScript程序

b. setTimeout是异步线程，需要等待js引擎处理完同步代码（while语句）之后才会执行，while语句直接是个死循环，js引擎没有空闲，不会执行下面的alert，也不会插入setTimeout。我在chrome中执行在线代码，最后浏览器是终止死循环执行alert。

c. JavaScript的工作机制是：当线程中没有执行任何同步代码的前提下才会执行异步代码，setTimeout是异步代码，所以setTimeout只能等js空闲才会执行，但死循环是永远不会空闲的，所以setTimeout也永远不会执行。

4.36、输出今天的日期，以YYYY-MM-DD的方式，比如今天是2014年9月26日，则输出2014-09-26

参考《JavaScript Date 对象》

4.37、Javascript中callee和caller的作用？

arguments.callee属性包含当前正在执行的函数。

Function.caller返回一个对函数的引用，该函数调用了当前函数。

参考资料

参考资料：

http://www.cnblogs.com/yexiaochai/p/3899974.html 【答阿里寒冬面试题】呵呵，大神的面试题就是好！

http://www.cnblogs.com/yexiaochai/p/3163657.html 做几道前端面试题休息休息吧

http://www.cnblogs.com/yexiaochai/p/3154031.html 来看一点CSS相关的吧

http://www.cnblogs.com/yexiaochai/p/3152858.html 前端面试题第二弹袭来，接招！

http://www.w3cfuns.com/blog-5449691-5399942.html 阿里巴巴校招笔试题整理（HTML+CSS篇）

http://blog.csdn.net/kongjiea/article/details/46341575 最全前端面试问题及答案总结

http://segmentfault.com/a/1190000000465431 2014年最新前端开发面试题

http://www.cnblogs.com/QingFlye/p/4295417.html 2014PPTV-题解

http://www.cnblogs.com/leolai/archive/2013/04/29/3050908.html 一些前端开发的笔试题及答案【编程题】

http://www.cnblogs.com/jscode/archive/2012/07/10/2583856.html 常见前端面试题之HTML/CSS部分

http://www.cnblogs.com/coco1s/p/4034937.html BAT及各大互联网公司2014前端笔试面试题--Html,Css篇

http://www.cnblogs.com/coco1s/p/4029708.html BAT及各大互联网公司2014前端笔试面试题--JavaScript篇【编程题】