web前端面试题

**其他问题**

1、请谈谈你对性能优化的认识？

网页内容

减少http请求次数

80%的响应时间花在下载网页内容(images, stylesheets, javascripts, scripts, flash等)。减少请求次数是缩短响应时间的关键！可以通过简化页面设计来减少请求次数，但页面内容较多可以采用以下技巧。

图片显示效果如左图

减少DNS查询次数

DNS查询也消耗响应时间，如果我们的网页内容来自各个不同的domain (比如嵌入了开放广告，引用了外部图片或脚本)，那么客户端首次解析这些domain也需要消耗一定的时间。DNS查询结果缓存在本地系统和浏览器中一段时间，所以DNS查询一般是对首次访问响应速度有所影响。下面是我清空本地dns后访问博客园主页dns的查询请求。

缓存Ajax

Ajax可以帮助我们异步的下载网页内容，但是有些网页内容即使是异步的，用户还是在等待它的返回结果，例如ajax的返回是用户联系人的下拉列表。所以我们还是要注意尽量应用以下规则提高ajax的响应速度。

延迟加载

这里讨论延迟加载需要我们知道我们的网页最初加载需要的最小内容集是什么。剩下的内容就可以推到延迟加载的集合中。

Javascript是典型的可以延迟加载内容。一个比较激进的做法是开发网页时先确保网页在没有Javascript的时候也可以基本工作，然后通过延迟加载脚本来完成一些高级的功能。

延迟加载

这里讨论延迟加载需要我们知道我们的网页最初加载需要的最小内容集是什么。剩下的内容就可以推到延迟加载的集合中。

Javascript是典型的可以延迟加载内容。一个比较激进的做法是开发网页时先确保网页在没有Javascript的时候也可以基本工作，然后通过延迟加载脚本来完成一些高级的功能。

2、如何避免XSS？

禁止危险脚本

IE8是第一款内置了XSS脚本拦截保护的浏览器。谷歌的Chrome也会紧随其后推出类似功能。这两款浏览器都会首先查看来自某个Web服务器的脚本是否是恶意的——如果是，就拦截它。在今年4月的黑帽欧洲2010大会上，研究专家David Lindsay和Eduardo Vela Nava却演示了一种可以破除这种拦截的办法，不过谷歌已经修复了Chrome中的这个漏洞。微软则在今年1月（补丁MS10-002）和3月（MS10-018）也已解决了大部分问题，并计划在6月修复第3个漏洞，所以在你读到这篇文章的时候，破除XSS脚本拦截的问题可能已经完全解决了。

Firefox的用户则可以利用免费的NoScrpit附加组件有选择地拦截脚本。比如说，你可以放行一段Flash视频，而同时拦截该网站上的其他脚本组件。IE和Chrome在拦截可疑脚本方面没有这么细的粒度——它们是要么全拦截，要么全不拦截。

NoScrpit也有一个问题，那就是大多数用户并不喜欢放行个别脚本的做法，因为这样会带来不便。不过拦截和放行今后可能会成为你的第二天性。你还可以对某个特定网站上的所有脚本进行认证，无论是为了一次性访问还是今后的所有访问，这样的认证如今在IE 8和chrome中也可以做了，使得防范XSS攻击实现更加可能。

3、平时如何管理项目？

所谓项目，简单地说，就是在既定的资源和要求的约束下，为实现某种目的而相互联系的一次性工作任务。一般来说，项目具有如下的基本特征：

1）明确的目标其结果只可能是一种期望的产品，也可能是一种所希望得到的服务。

2）独特的性质每一个项目都是唯一的。

3）资源成本的约束性每一项目都需要运用各种资源来实施，而资源是有限的。

4）项目实施的一次性项目不能重复。

5）项目的不确定性在项目的具体实施中，外部和内部因素总是会发生一些变化，因此项目也会出现不确定性。

4、请谈谈项目的迭代周期？

软件项目开发，一般都会采用增量、迭代、（或者叫进化、演化、演进）的软件开发模型，众多的软件开发模型大多是以经典的瀑布模型为基础进行改进、变形，改进原则是：增加客户在整个项目周期中的参与度，降低软件开发过程中的风险，增强软件项目的后期可维护性。

不同的软件开发模型，迭代周期长短也不相同，有的是一个月，有的是两周，我们一般都是根据实际情况确定，一个周期完成，将项目成果（可运行的软件）提交给用户（或进行内部评审），通过后就进入下一个迭代开发周期

5、工作中用过什么构建工具？

答：用过gulp。

第一步：安装node和npm,搭建node环境。

第二步：安装gulp

第三步：新建Gulpfile文件，运行gulp

安装依赖，提醒下，如果以上命令提示权限错误，需要添加 sudo 再次尝试。

Gruntfile维护起来那么困难，有几个原因：

配置和运行分离

程序员都知道，变量的声明和使用挨在一起，最方便理解和修改。但Gruntfile里，配置Task和调用它们的地方离得很远，极大地增加了心智负担

每个插件做的事太多

每个Task的结果必须写到磁盘文件，另一个Task再读，损害性能倒是小事，更麻烦的是让整个过程变复杂了。就像一个个小作坊，来料加工又返回给客户，这中间的沟通成本、出错机会都大大增加。配置项过多做事多了，配置项自然也多。至少输入和输出的位置得配吧。每个插件的配置规则还不尽相同。用每个插件，都得去学习一番。

6、谈谈你对模块化的理解？

什么是模块化？

• 模块化就是为了减少系统耦合度，提高高内聚，减少资源循环依赖，增强系统框架设计。

• 让开发者便于维护,同时也让逻辑相同的部分可复用

• 模块化开发：针对js、css，以功能或业务为单元组织代码。js方面解决独立作用域、依赖管理、

api暴露、按需加载与执行、安全合并等问题，css方面解决依赖管理、组件内部样式管理等问题。

任何事物都有一个过程，那么模块化的过程通俗点讲就是：

模块化的过程就是：

• 1、拆分

将整个系统按功能,格式,加载顺序,继承关系分割为一个一个单独的部分.

注意:拆分的粒度问题,可复用问题,效率问题.如何这些问题处理的不好，就有可能出现不想要的后果。

将功能或特征相似的部分组合在一起,组成一个资源块.

将每个资源块按找需求,功能场景以及目录约束放到固定的地方以供调用.

模块的历程

模块化的发展也是从草根一步一步走过来的。从最开始到现在成熟方案：

1. namespace

2. sass,less

3. AMD&CMD

4. html模版

5. grunt,gulp,webpack

6. FIS：FIS3采取了类似CSS语法一样的配置风格,

YUI：YUI库是一系列使用Javascript和CSS创建的的工具和控件集,

KISSY：KISSY 是一个框架，一个生态系统，里面不仅有DOM操作，还有什么模块加载功能

7、平时都用什么第三方框架？

jQuery、Bootstrap、angularjs、vue、underscore、seajs、RequireJS…

1. 请描述一下cookie，sessionStorage和localStorage的区别？

答：cookies兼容所有的浏览器，Html5提供的storage存储方式。

* Document.cookie
* Window.localstorage
* Window.sessionstorage

cookie数据始终在同源的[http请求](https://www.baidu.com/s?wd=http%E8%AF%B7%E6%B1%82&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3Phndmy7buj64mWPBmhFh0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1b1n1bsnjTvrjnzPjcvnHmd)中携带（即使不需要），即cookie在浏览器和服务器间来回传递。而sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。

存储大小限制也不同，cookie数据不能超过4k，同时因为每次[http请求](https://www.baidu.com/s?wd=http%E8%AF%B7%E6%B1%82&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3Phndmy7buj64mWPBmhFh0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En1b1n1bsnjTvrjnzPjcvnHmd)都会携带cookie，所以cookie只适合保存很小的数据，如会话标识。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。

数据有效期不同，sessionStorage：仅在当前浏览器窗口关闭前有效，自然也就不可能持久保持；localStorage：始终有效，窗口或浏览器关闭也一直保存，因此用作持久数据；cookie只在设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭。

作用域不同，sessionStorage不在不同的浏览器窗口中共享，即使是同一个页面；localStorage 在所有同源窗口中都是共享的；cookie也是在所有同源窗口中都是共享的。

9、如何使用缓存？

可以基于http的头信息控制缓存

ajax请求对早期的IE浏览器默认就是缓存的，可以通过时间戳防止缓存

10、谈谈你对预加载的理解？

Web预加载指的是在网页全加载完成之前，在页面优先显示一些主要内容，以提高用户体验。对于一个较庞大的网站，如果没有使用预加载技术，用户界面就会长时间显示一片空白，直到资源加载完成，页面才会显示内容。

例如，可以通过js预先从服务加载图片资源(动态创建Image，设置src属性)，只要浏览器把图片下载到本地，就会被缓存，再次请求相当的src时就会优先寻找浏览器缓存，提高访问速度。

11、缓存和预加载的区别是什么？

缓存就是把请求过的数据缓存起来，下次请求的时候直接使用缓存内容，提高响应速度

预加载指的是提前把需要的内容加载完成，访问的时候可以明天提高响应效率，比如图片的预加载（可以提前加载一定数量的图片，当用户访问图片的时候一般只看前几张，由于是预加载好的，所以速度比较快）

12、图片如何压缩？

可以使用一些在线的图片压缩工具

优先用 png 而不是 gif

压缩 png

去掉 jpg 的 metadata

压缩 gif 动画

尝试使用 png8

避免使用 AlphaImageLoader

压缩动态生成的图像

使 favicon 更小 可缓存

使用 CSS Sprites

13、压缩文件有哪些方法？

使用Grunt：自动化。对于需要反复重复的任务，例如压缩（minification）、编译、单元测试、Sass、

ant压缩

14、如何区分静态页面和动态页面？

答：要区分这两个，最简单的方法就是看后缀了，动态网页网址中有两个标志性的符号“?”和“&”（有的可能没有&），这个问号和&就是用来带参数的。现在几乎爱所有的网页都是动态网页。

15、字符串拼接和模板引擎，项目中会如何操作？模板引擎会不会利于SEO优化？

简单的数据渲染，拼接字符串即可，稍微复杂的业务逻辑使用前端模板引擎，过于复杂的页面基本上使用后台渲染的方式；模板引擎会影响SEO优化，为了解决这个问题，需要关注SEO的页面最好采用后台渲染的方式。

16、前台兼容性问题有哪些？

答：主要是常用浏览的（前端）API差异，渲染差异，等等

17、你如何对网站的文件和资源进行优化?期待的解决方案包括？

答：文件合并

文件最小化/文件压缩

使用CDN托管

缓存的使用

18、内存泄漏怎么理解？

答： IE6时代有bug，闭包会造成内存泄漏，这个现在已经无须考虑了。  
其次，闭包本身不会造成内存泄漏，但闭包过多很容易导致内存泄漏。  
这句话很矛盾，技术上讲，闭包是不会造成内存泄漏的，浏览器的bug除外。但是，闭包会造成对象引用的生命周期脱离当前函数的上下文，因此，如果不仔细考虑闭包函数的生命周期，的确有可能出现意料之外的内存泄漏，当然，从严格意义上讲，这是程序员自己的bug，而不是闭包的错。

19、微格式到底是做啥用？

答：是开放的数据格式，面向的是普通用户，任何用户可以透过简单的程序读取微格式内容。而不是像Flickr、Amazon、Google等提供特定的面向技术人员的API（一般基于XML-PRC、REST，相对复杂）。RSS具有微格式的部分优点，但限制还是比较多的，比如有限的元数据（标题、描述、URL等），不能更好地描述语义，不太容易与已存在的工具结合等。用微格式可以来聚合外部Blog，Flickr，YouTube，MapQuest，甚至MySpace里的内容。

微格式实际就是为现有的(X)HTML元素添加元数据和其他属性，增强语义。

20、懒加载是用滚轮判断高度好还是用插件？

使用插件比较好，插件考虑的问题比较全面，仅仅通过滚轮高度判断很容易导致一些副作用（比如一次性请求多次）

**21、如何缓存整个页面，在没有网络的时候可以来回的跳转？**

使用 HTML5，通过创建 cache manifest 文件，可以轻松地创建 web 应用的离线版本。

如需启用应用程序缓存，请在文档的 <html> 标签中包含 manifest 属性。

每个指定了 manifest 的页面在用户对其访问时都会被缓存。如果未指定 manifest 属性，则页面不会被缓存（除非在 manifest 文件中直接指定了该页面）。

manifest 文件的建议的文件扩展名是：".appcache"。

manifest 文件需要配置正确的 MIME-type，即 "text/cache-manifest"。必须在 web 服务器上进行配置。

manifest 文件可分为三个部分：

CACHE MANIFEST - 在此标题下列出的文件将在首次下载后进行缓存

NETWORK - 在此标题下列出的文件需要与服务器的连接，且不会被缓存

FALLBACK - 在此标题下列出的文件规定当页面无法访问时的回退页面（比如 404 页面）

*实例 - 完整的 Manifest 文件*

CACHE MANIFEST

# 2012-02-21 v1.0.0

/theme.css

/logo.gif

/main.js

NETWORK:

login.asp

FALLBACK:

/html5/ /404.html

**22、CDN是啥？**

CDN的全称：是Content Delivery Network，即内容分发网络，加速的意思，那么网站CND服务就是网站加速服务。

CDN加速原理：CDN加速将网站的内容缓存在网络边缘（离用户接入网络最近的地方），然后在用户访问网站内容的时候，通过调度系统将用户的请求路由或者引导到离用户接入网络最近或者访问效果最佳的缓存服务器上，有该缓存服务器为用户提供内容服务；相对于直接访问源站，这种方式缩短了用户和内容之间的网络距离，从而达到加速的效果。

CDN的特点：

1、本地加速 提高了企业站点(尤其含有大量图片和静态页面站点)的访问速度，并大大提高以上性质站点的稳定性

2、镜像服务 消除了不同运营商之间互联的瓶颈造成的影响，实现了跨运营商的网络加速，保证不同网络中的用户都能得到良好的访问质量。

3、远程加速 远程访问用户根据DNS负载均衡技术 智能自动选择Cache服务器，选择最快的Cache服务器，加快远程访问的速度

4、带宽优化 自动生成服务器的远程Mirror(镜像)cache服务器，远程用户访问时从cache服务器上读取数据，减少远程访问的带宽、分担网络流量、减轻原站点WEB服务器负载等功能。

5、集群抗攻击 广泛分布的CDN节点加上节点之间的智能冗于机制，可以有效地预防黑客入侵以及降低各种D.D.o.S攻击对网站的影响，同时保证较好的服务质量。

**23、浏览器一次可以从一个域名下请求多少资源？**

浏览器的并发请求数目限制是针对同一域名的，同一时间针对同一域名下的请求有一定数量限制，不同浏览器这个限制的数目不一样，超过限制数目的请求会被阻塞；

目前的话，所有浏览器的并发数目一般限制在10以内。

**24、什么是垃圾回收机制（GC）？**

早期的计算机语言，比如C和C++，需要开发者手动的来跟踪内存，这种机制的优点是内存分配和释放的效率很高。但是它也有着它的缺点，程序员很容易不小心忘记释放内存，从而造成内存的泄露。

新的编程语言，比如JAVA, C#， javascript, 都提供了所谓“垃圾回收的机制”，运行时自身会运行相应的垃圾回收机制。程序员只需要申请内存，而不需要关注内存的释放。垃圾回收器(GC)会在适当的时候将已经终止生命周期的变量的内存给释放掉。GC的优点就在于它大大简化了应用层开发的复杂度，降低了内存泄露的风险。

**25、image和canvas在处理图片的时候有什么区别？**

image是通过对象的形式描述图片的

canvas通过专门的API将图片绘制在画布上

**26、简述移动开发的注意点,如何做好不同手机的适配,你以前的项目是怎么做的?**

1、单独做移动端项目，采用百分比布局

2、采用响应式的方式做适配

**27、响应式布局的时候，轮播图使用两张不同的图片去适配大屏幕和超小屏幕，还是一张图片进行压缩适配不同终端，说明原因？**

最好使用两张不同大小的图片去适配大屏幕和超小屏幕，这样可以针对不同设备的屏幕大小，来加载响应的图片，减少超小屏幕设备的网络流量消耗，加快响应速度，同时防止图片在大屏幕下分辨率不够导致失真的问题。

**28、http和tcp有什么区别？**

TPC/IP协议是传输层协议，主要解决数据如何在网络中传输，是一种“经过三次握手”的可靠的传输方式；

HTTP协议即超文本传送协议(Hypertext Transfer Protocol )，是应用层协议，是Web联网的基础，也是手机联网常用的协议之一，HTTP协议是建立在TCP协议之上的一种应用。

**29、向git中添加一个文件并commit，然后push到remote server,请写出相关命令？**

$ git add README.md

$git commit -m "add README.md"

$ git push origin master

**30、请把以下HTML文档翻译成MarkDown格式？**

<h3>Header</h3>

<p>Hello world!<a href="https://www.google.com">Google</a></p>

<ol>

<li>Number One</li>

<li>Number Two</li>

</ol>

### Header

Hello world![Google](https://www.google.com)

1. Number One

2. Number Two

**35、设计模式有哪些？列举你在前端开发工作中自己应用到或者了解到其他框架所用到的设计模式？**

单例、工厂、观察者、适配器、代理模式

**36、请描述你熟悉的语言的垃圾回收(GC)机制，他们对循环引用是如何处理的？如何查找内存泄漏(MemoryLeak)?**

JavaScript的垃圾回收机制主要是根据数据是否还存在引用，没有引用的数据空间可能在某个时间被回收；在java中垃圾回收机制采用对象遍历来解决循环引用；windows的任务管理器就可以查看到内存泄露

31、你做的页面在哪些流览器测试过?这些浏览器的内核分别是什么?

答：Ie(Ie内核) 火狐（Gecko）谷歌（webkit） opear(Presto)

32、写出几种IE6 BUG的解决方法？

答：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dislpay: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.Ie 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）

33、图片优化

项目中图片处理相关的优化，项目中用到的优化方案，图片大小达到多少的时候选择处理？

答：1、首先了解在web开发中常见的图片有那些格式。

JPG 通常使用的背景图片，照片图片，商品图片等等。这一类型的图片都属于大尺寸图片或较大尺寸图片一般使用的是这种格式。

PNG 这种格式的又分为两种一种PNG-8，一种 PNG-24。

PNG-8格式不支持半透明，也是IE6兼容的图片存储方式。

PNG-24图片质量要求较高的半透明或全透明背景，保存成PNG-24更合适（为了兼容IE可以试用js插件pngfix）一般是背景图标中使用的多。

GIF 这种格式显而易见的是在需要gif动画的时候使用了。

2．优化方案

* 样式代替图片

例如：半透明、圆角、阴影、高光、渐变等。这些效果主流的浏览器都能够完美支持，而对于那些低端浏览器，我们并不会完全抛弃他们，“**渐进增强”**则是一个很好的解决方案。

* 精灵图

CSS Sprites，将同类型的图标或按钮等背景图合到一张大图中，减少页面请求。

* 字体图标

Icon Font，将图标做成字体文件。优点是图标支持多个尺寸，兼容所有浏览器，减少页面请求等。美中不足的是只支持纯色的icon。SVG，对于绝大多数图案、图标等，矢量图更小，且可缩放而无需生成多套图。现在主流浏览器都支持SVG了，所以可放心使用！

* Base64

将图片转化为base64编码格式，资源内嵌于CSS或HTML中，不必单独请求。

Base64格式

data:[][;charset=][;base64],

Base64 在CSS中的使用

.demoImg{ background-image: url("data:image/jpg;base64,/9j/4QMZRXhpZgAASUkqAAgAAAA L...."); }

Base64 在HTML中的使用

<img width="40" height="30" src="data:image/jpg;base64,/9j/4QMZRXhpZgAASUkqAAgAAAAL...." />

* 图片响应式

通常图片加载都是可以通过lazy加载的形式来的，那么可以在加载的时候来判断屏幕的尺寸来达到加载大图还是小图的目的来达到优化。

34、你知道有哪些方法可以提高网站的性能？

答：我们从两个方面来讲：

1. 资源加载

CSS顶部, JS底部

CSS JS文件压缩

尽量使用图片使用精灵图，字体图标

图片加载可通过懒加载的方式。

总之就是减少资源体积减少资源请求次数。

1. 代码性能

Css:

使用CSS缩写，减少代码量；

减少查询层级：如.header .logo要好过.header .top .logo；

减少查询范围：如.header>li要好过.header li；

避免TAG标签与CLASS或ID并存：如a.top、button#submit；

删除重复的CSS；

….

Html:

减少DOM节点：加速页面渲染；

正确的闭合标签：如避免使用<div/>，浏览器会多一个将它解析成<div\></div\>的过程；

减少页面重绘。比如给图片加上正确的宽高值：这可以减少页面重绘，

……

Js：

尽量少用全局变量；

使用事件代理绑定事件，如将事件绑定在body上进行代理；

避免频繁操作DOM节点；

减少对象查找，如a.b.c.d这种查找方式非常耗性能，尽可能把它定义在变量里；

**Angular、主流框架和服务器相关问题**

1. ng-app是什么？

答：

**ng-app** 指令用于告诉 AngularJS 应用当前这个元素是根元素。

所有 AngularJS 应用都必须要要一个根元素。

HTML 文档中只允许有一个 ng-app 指令，如果有多个 ng-app 指令，则只有第一个会被使用。

2、说说MVC和MVVM分别是什么？

MVC全名是Model View 。是Controller模型(model)－视图(view)－控制器(controller)的缩写

*MVVM*是Model-View-ViewModel的简写

3、−g是什么？

在Node.js中-g表示全局安装模块

4、自定义指令的类型（E，A，C，M）？

元素（E）、属性（A）、类（C）、注释（M）

5、$scope和自定义指令里的scope有啥区别？

$scope 对象在 AngularJS 中充当数据模型的作用,也就是一般 MVC 框架中 Model 得角色.但又不完全与通常意义上的数据模型一样,因为 $scope 并不处理和操作数据,它只是建立了视图和 HTML 之间的桥梁,让视图和Controller 之间可以友好的通讯.

自定义指令里的scope表示指令的作用域，它有三个可选值：true、false、对象{}

6、Ionic中的路由？

Ionic也是基于Angular的，使用的是ui-router,

ui-router的核心理念是将子视图集合抽象为一个状态机，导航意味着 状态的切换,Ionic之所以没有使用Angular官方的ngRoute,是回ngRoute缺少一些高级的特性，比如视图命名,视图嵌套。

7、filter？

过滤器

8、ng-bind？

ng-bind 指令告诉 AngularJS 使用给定的变量或表达式的值来替换 HTML 元素的内容。

如果给定的变量或表达式修改了，指定替换的 HTML 元素也会修改。

9、说一说link？

link中可以拿到scope和controller，可以与scope进行数据绑定，与其他指令进行通信。

10、为什么angular不推荐使用dom操作？

Angular倡导以测试驱动开发，在的service或者controller中出现了DOM操作，那么也就意味着的测试是无法通过的

使用Angular的其中一个好处是啥，那就是双向数据绑定，这样就能专注于处理业务逻辑，无需关系一堆堆的DOM操作。如果在Angular的代码中还到处充斥着各种DOM操作，那为什么不直接使用jquery去开发呢

11、看过Angular的源码吗，它是怎么实现双向数据绑定的？

angular对常用的dom事件，xhr事件等做了封装， 在里面触发进入angular的digest流程。  
在digest流程里面， 会从rootscope开始遍历， 检查所有的watcher

12、讲讲promise:{  
  promise:null;;  
｝？

13、ui-router 和 ng-router区别？

AngularJS的ng-route模块为[控制器](https://www.baidu.com/s?wd=%E6%8E%A7%E5%88%B6%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdujmsPHRsmWfvPyDdnHbd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En10Lnjfdnj6vPW03PjDsnW0Y)和视图提供了[Deep-Linking]URL

ui-router的核心理念是将子视图集合抽象为一个状态机，导航意味着 状态的切换。

14、什么是指令？

*指令*是指示计算机执行某种操作的命令,它由一串二进制数码组成。一条*指令*通常由两个部分组成:操作码+地址码。

15、service服务三种方式是什么？

angularjs 中可通过三种（$provider,$factory,$service）方式自定义服务

16、gulp任务都是怎么定义，怎么执行的？

通过 gulp.task方法定义任务,在项目中新建gulpfile.js文件，书写代码,如:

var gulp = require(‘gulp’)

gulp.tasks(

‘script’ // 任务名

,function(){

// 在这里写任务需要执行的代码

});

在命令输入`gulp 任务`,可以执行所在目录gulpfile.js文件中的任务。

17、Bootstrap中最多可以分多少列？lg、md、sm、xs这几个屏幕宽度的界限是多少？

12列

.col-xs- 超小屏幕手机 (<768px)  
.col-sm- 小屏幕平板 (≥768px)  
.col-md- 中等屏幕桌面显示器 (≥992px)  
.col-lg- 大屏幕大桌面显示器 (≥1200px)

18、angular中方法apply和digest区别？

当数据出现没有经过angular但是发生改变的情况下，需要调用apply。Apply的范围比较广，只执行一次，但是digest针对某一元素执行多次。

20、ng-show/hide和ng-if的区别是什么？

Show/hide是显示隐藏，if是是否存在某一部分

21、react虚拟DOM运行机制是什么？

在React中，render执行的结果得到的并不是真正的DOM节点，结果仅仅是轻量级的JavaScript对象，我们称之为virtual DOM。虚拟dom

22、react中prop和state的区别？

需要理解的是，props是一个父组件传递给子组件的数据流，这个数据流可以一直传递到子孙组件。而state代表的是一个组件内部自身的状态（可以是父组件、子孙组件）。

23、redux的原理？

Redux 把一个应用程序中，所有应用模块之间需要共享访问的数据，都应该放在 State 对象中。这个应用模块可能是指 React Components，也可能是你自己访问 AJAX API 的代理模块，具体是什么并没有一定的限制。State 以 “树形” 的方式保存应用程序的不同部分的数据。这些数据可能来自于网络调用、本地数据库查询、甚至包括当前某个 UI 组件的临时执行状态（只要是需要被不同模块访问）、

24、node常用模块？

http fs path url Buffer process

25、了解npm，spm，nodejs吗，请简要描述？

NPM便于JavaScript开发者共享和重用代码，它可以很容易地更新你的代码；再分享。是全球最大的开源库生态系统。

SPM是淘宝社区电商业务（xTao）为外部合作伙伴（外站）提供的一套跟踪引导成交效果数据的解决方案。

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。Node.js 使用了一个事件驱动、非阻塞式 I/O 的模型，使其轻量又高效。它使我们能够在本地运行javascript

26.请列举在内网的两台服务器中拷贝文件的方法？用Shell脚本解答数据库？

scp -P 1234 -r test\_folder user@192.168.1.20:~

主要就是scp命令的使用

28、请描述你所熟悉的Web服务器框架(如Django)作为一个成熟的Web框架，需要提供哪些重要的功能模块？

提供了网站开发的常用模块：处理用户请求、操作数据库、模板渲染、配置文件管理等

29、服务器Node.js和浏览器js的区别是什么？Node.js把js从客户端迁移了到服务端、主要做了哪些工作？为什么说Node.js适合做高并发的互联网应用？

Node采用一系列“非阻塞”库来支持事件循环的方式。本质上就是为文件系统、数据库之类的资源提供接口。Node.js 使用事件驱动，非阻塞I/O 模型而得以轻量和高效，非常适合在分布式设备上运行数据密集型的实时应用。

**网络相关问题**

1、请解释下列术语：UrlEncode,Utf8,JSON,UTC,MD5？

Urlencode：将字符串以url形式编码（在编程语言中通常都有实现该功能的内置函数或者API）。

Utf8：是一种针对Unicode的可变长度字符编码，主要用于在网页上显示各国语言字符

2、请解释GET/POST的区别，以及请求参数放到url里和放到body里面的区别？

Post与Get区别：

GET请求，请求的数据会附加在URL之后，以?分割URL和传输数据，多个参数用&连接。URL的编码格式采用的是ASCII编码，而不是uniclde，即是说所有的非ASCII字符都要编码之后再传输。

POST请求：POST请求会把请求的数据放置在HTTP请求包的包体中。上面的item=bandsaw就是实际的传输数据。

因此，GET请求的数据会暴露在地址栏中，而POST请求则不会。

传输数据的大小

在HTTP规范中，没有对URL的长度和传输的数据大小进行限制。但是在实际开发过程中，对于GET，特定的浏览器和服务器对URL的长度有限制。因此，在使用GET请求时，传输数据会受到URL长度的限制。

对于POST，由于不是URL传值，理论上是不会受限制的，但是实际上各个服务器会规定对POST提交数据大小进行限制，Apache、IIS都有各自的配置。

安全性

Get是Form的默认方法，安全性相对比较低。

请求参数放到url里和放到body里面的区别

首先，参数的存放位置我们无法直接指定，而是不同的请求方法参数传递的方式不同。

常用的HTTP请求主要为GET请求和POST请求两种，GET请求的参数会通过以跟随在URL后边以键值对的方式进行传递（例：key1=a&key2=b&key3...)；而POST请求的参数会通过HEADER进行传递。考虑到安全性的问题，可以确定两者都不安全，原因是HTTP请求可以被轻易抓包和截获，其中的请求参数值自然会很容易被获取。

3、请列举出常用的Http Header，Cookie是怎么实现的？

Content-Length，请求、响应体的数据字节大小  
Accept-Encoding，请求头，可接受的文本压缩算法，如： gzip, deflate  
Accept-Language，请求头，支持语言，客户端浏览器的设置，如：zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3  
User-Agent，请求头，浏览器信息，

Cookie，请求头，服务器或客户端在上次设置的COOKIE，包括作用域名(.360buy.com)，过期时间，键与值。

Referer，从一个连接打开一个新页面，新页面的请求一般会加此信息，标名是从哪里跳过来的，所有的页面的打开历史链就可被挖掘出来，有利于分析用户行为与CPS分成

Cookie在浏览器本地会有一个文件存储数据，通信的时候通过请求头和响应头传递数据

4、请解释下列返回码的含义：200,302,400,403,500,502

200：请求成功

302：请求的资源临时从不同的 URI响应请求。(资源临时重定向)

400：错误请求（请求的参数错误或者服务器不理解请求的语法）

402：10.4.3 402 Payment Required This code is reserved for future use.

该状态码是为了将来可能的需求而预留的。

500：服务器端错误

502：网关或代理无效/无响应，网络错误

5、长连接和短连接的区别

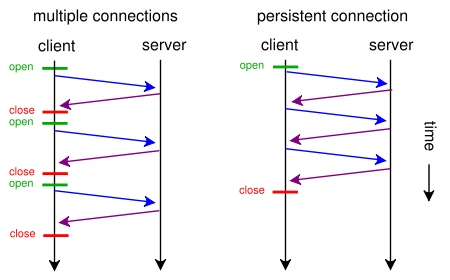
HTTP协议目前常用的有哪几个？KEEPALIVE从哪个版本开始出现的？

到现在http出现了 1.0和 1.1版本

Keep-Alive是从1.1默认就支持了。

1、什么是Keep-Alive模式？

我们知道HTTP协议采用“请求-应答”模式，当使用普通模式，即非KeepAlive模式时，每个请求/应答客户和服务器都要新建一个连接，完成之后立即断开连接（HTTP协议为无连接的协议）；当使用Keep-Alive模式（又称持久连接、连接重用）时，Keep-Alive功能使客户端到服务器端的连接持续有效，当出现对服务器的后继请求时，Keep-Alive功能避免了建立或者重新建立连接。



http 1.0中默认是关闭的，需要在http头加入"Connection: Keep-Alive"，才能启用Keep-Alive；http 1.1中默认启用Keep-Alive，如果加入"Connection: close "，才关闭。目前大部分浏览器都是用http1.1协议，也就是说默认都会发起Keep-Alive的连接请求了，所以是否能完成一个完整的Keep-Alive连接就看服务器设置情况。

2、启用Keep-Alive的优点

从上面的分析来看，启用Keep-Alive模式肯定更高效，性能更高。因为避免了建立/释放连接的开销。

6、从服务器考虑提高网站性能

答：业界常用的优化WEB页面加载速度的方法（可以分别从页面元素展现，请求连接，css,js,服务器等方面介绍）？

对于服务器方面前端能做的工作：

使用CDN加速，使用户从离自己最近的服务器下载文件；

减少Cookie的大小，使用无cookie的域，客户端请求静态文件的时候，减少 Cookie 的反复传输对主域名的影响；

为文件头指定Expires，使内容具有缓存性；

前端优化:DNS预解析提升页面速度

<link rel="dns-prefetch" href="http://hm.baidu.com" />

<link rel="dns-prefetch" href="http://eiv.baidu.com" />

服务器端能做的工作：

负载均衡，分布式存储，提升服务器性能等等。

1. 什么是Daemon进程？

答：Daemon()程序是一直运行的服务端[程序](http://baike.baidu.com/subview/17674/13521310.htm)，又称为[守护进程](http://baike.baidu.com/view/53123.htm)。通常在系统后台运行，没有控制终端，不与前台交互，Daemon程序一般作为[系统服务](http://baike.baidu.com/view/685551.htm)使用。Daemon是长时间运行的进程，通常在系统启动后就运行，在系统关闭时才结束。一般说Daemon程序在后台运行，是因为它没有控制终端，无法和前台的用户交互。Daemon程序一般都作为服务程序使用，等待客户端程序与它通信。我们也把运行的Daemon程序称作守护进程。

8、优化一个以I/O为瓶颈的程序，以下哪些方法效果比较显著，Why?

* 1. 增加CPU数目
  2. 提高CPU主频
  3. 增大内存的容量
  4. 采用多线程
  5. 采用异步I/O和多路(Multlplex)I/O
  6. 对每次I/O进行Bath访问(多次I/O合并一次完成)

c、e、d、f提升的效果会比较显著

c通过将数据预读取到内存中（建立内存池）的方式，提高访问时候的效率，有效减少磁盘IO读写次数

9、设计一个系统，实现统计网站中每个URL访问到的次数(pv)，以及访问的独立用户数(uv)，其中每个独立网页以URL为key，每个独立用户以ip为key？

10、什么是内存对象的序列化(Serialiization)？为什么要序列化？请描述你熟悉的网络传输序列化(Serialiization)框架或格式(Server)？

把对象转换为字节序列的过程称为对象的序列化

序列化主要用于网络传输数据及将数据保存在硬盘上

常见的序列化以后的格式有：XML Jason ，但它们都是字符串

**项目相关问题**

1、请谈下团购倒计时如何实现？

团购倒计时页面端的效果比较好实现，主要是样式和时间的操作，重要的考虑时间要和服务器端同步，其实这个效果也可以基于服务器端推送技术来实现

2、轮播图有哪几种？如何实现？

纯css可以实现轮播图；js实现轮播图

3、如何实现数组去重？

a.最简单的可以直接利用ES5的indexOf方法。

function dupRemove(arr) {

var tmp = []; //一个新的临时数组

for (var i = 0; i < arr.length; i++) {

// 判断tmp数组中是否存在arr中第i元素，如果不存在则添加到tmp数据组。

if (tmp.indexOf(arr[i]) == -1) {

tmp.push(arr[i]);

}

}

return tmp;

}

b.

b.还有一种比较有意思的写法  
function dupRemove (arr) {

var tmp = [];

for (var i = 0, l = arr.length; i < l; i++) {

for (var j = i + 1; j < l; j++) {

if (arr[i] === arr[j]) {

j = ++i;

}

}

tmp.push(arr[i]);

}

return r;

}

4、写一个方法获取url中？号后面的参数，并将参数对象化?  
function getSearch (url) {

var reg\_url = /^[^\?]+\?([\w\W]+)$/,

reg\_params = /([^&=]+)=([\w\W]\*?)(&|$|#)/g,

arr\_url = reg\_url.exec(url),

ret = {};

if (arr\_url && arr\_url[1]) {

var str\_ params = arr\_url[1], result;

while ((result = reg\_ params.exec(str\_ params)) != null) {

ret[result[1]] = result[2];

}

}

return ret;

}

**程序题**

1、var a=[];

a[0]=0;a[1]=1;a[4]=4;

请问a.length的值是多少？a[3]的输出结果是什么？

**5 undefined**

2、var a=[5,6];

var b=a;

b[0]="hello";

alert(a[0]);

请问值是多少？

**"hello"**

3、typeof(null),typeof(undefined),typeof(NaN),typeof(NaN==NaN),说出上面代码执行结果？

**object**

**undefined**

**number**

**boolean**

4、function doSomething(){

for(var i = 0; 4 > i; i++) {

var k = 100;

aMrg +=’,’ + (k + i);

}

}

var k = 1,aMrg = k;

doSomething();

aMrg +=k;

log(aMrg);

**1,100,101,102,1031**

5、请写出下面输出的值

Console.log(undefined || 1);//值\_\_\_**1**\_\_

Console.log(null || NaN);//值\_\_**NaN**\_\_\_

Console.log(0 && 1);//值\_**\_0**\_\_\_

Console.log(0 && 1 || 0);//值\_\_**0**\_\_\_

6、看下列代码，<p>标签内的文字是什么颜色的？**红色**

<style>

.classA{color: blue};

.classB{color: red};

</style>

<body>

<p class=”classB classA”>123</p>

</body>

7、var a = [5,6];

var b = a;

b[0] = “hello”;

alert(a[0]);

值是多少？

**"hello"**

1. 你面前有一座高塔，这座高塔有N(N > 100)个台阶，你每次只能往前迈1个或者2个台阶，请写出程序计算总共有多少种走法？

**这个案例满足斐波那契定律 1，1，2，3，5，8，13，21, 34, 55, 89, 144**

varn1 = 1;  
var n2 = 1;  
var n3 = n1 + n2;

for (var i = 3; i <= n; i++) {  
n3 = n1 + n2;  
n1 = n2;//往后推一项  
n2 = n3;//往后推一项  
}  
console.log(n3);

9、请阅读下面的CSS代码

#left {

color: white !important;

}

#container #left {

color: red;

}

#left {

color: green !important;

}

.container #left {

color: blue;

}

则在如下html中

<div class=”container” id=”container”>

<span id=”left”>left</span>

</div>

#left最终color属性值为？**绿色**

10、下面这段代码想要循环延时输出结果0 1 2 3 4，请问输出结果是否正确，如果不正确说明为什么，并修改循环内的代码使其输出正确的结果。

for (var i = 0; i < 5; ++i) {

setTimeout (function () {

console.log(i + ‘’);

},100\*i);

}

**不正确，先执行FOR循环。for循环完成后，在去执行setTimeout。但是这个时候I已经是5了，所以输入了5次5**

for(var i = 0; i <5; ++i) {  
var a = 0;  
setTimeout (function () {  
**console**.log(a++);  
 },100\*i);  
}

11、完成函数showlmg()，要求能够动态根据下拉列表的选项变化，更新图片的显示

<body>

<script type=”text/javascript”>

Function showImg (oSel) {

};

</script>

<img id=”pic” src=”img1.jpg” width=”200” height=”200”>

<br />

<select id=”sel” onchange=”showImg(this)”>

<option value=”img1”>城市生活</option>

<option value=”img2”>都市早报</option>

<option value=”img3”>青山绿水</option>

</select>

</body>

varpic=**document**.getElementById('pic')  
function showImg (oSel) {  
pic.**src**=oSel.**options**[oSel.**selectedIndex**].**value  
 console**.log(pic.**src**);  
};

答案说明：当select发生改变的时候调用showImg函数，实参为this（select对象本身），可以通过select对象的属性来为pic的src赋值实现图片切换

12、完成foo()函数的内容，要求能弹出对话框提示当前选中的是第几个单选框

<html>

<head>

<meat http-equiv=”Content-Type” content=”text/html; charset=utf-8”>

</head>

<body>

<script type=”text/javascript”>

function foo() {

};

</script>

<form name=”form1” onsubmit=”retuen foo()”>

<input type=”radio” name = “radioGroup”>

<input type=”radio” name = “radioGroup”>

<input type=”radio” name = “radioGroup”>

<input type=”radio” name = “radioGroup”>

</form>

</body>

</html>

var a=**document**.getElementsByTagName('input')  
function foo() {  
for(var i=0;i<a.**length**;i++){  
if(a[i].**checked**){  
alert(i+1)  
 }  
  
 }  
}

题粗的有问题，onsubmit只有在提交的时候才会触发这里面没有submit按钮，在提交事件触发的时候遍历哪个input表单是选中状态然后alert粗来

13、计算下面程序运行结果

var msg = ‘hello’;

function great(name, attr) {

name = ‘david’;

var greating = msg + name + ‘!’;

var msg = ‘您好’;

For (var i = 0 ; i < 10;i++) {

var next = msg + ‘您的id是’ + i\*2 + i;

}

console.log(arguments[0]);

console.log(arguments[1]);

console.log(greating);

console.log(next);

}

geat(‘Tom’);

答案：david //参数1

hellow world 01.html:20 undefined //参数2 未传入为未定义

hellow world 01.html:21 undefineddavid! //name虽然是参数但是参数重新赋值为david了msg因为变量声明提升所以值为undefined

您好您的id是189//因为number+string=string所以for循环最后一次声明next=\*\*\*\*18+9

14、下面这段JS输出什么，并简述为什么？

function Foo() {

var i = 0;

return function () {

console.log(i++);

}

}

var f1 = Foo(),

f2 = Foo();

f1();

f1();

f2();

console.log(i);

0 //f1=Foo() 相当于f1赋值为函数Foo()的返回值f1=function（）{

console.log(i++)

}

1 //因为f1=了一个function所以有了作用域，f2和f1不同，不在一个内存中

0

报错 //i为Foo内部的变量全局不可访问，全局中没有i变量所以会报错

15、请写出下面输出的值

* 1. var num = 1;

var fun = function () {

console.log(num);//值\_\_\_undefined\_\_\_

var num = 2;

console.log(num);//值\_\_\_2\_\_\_

}

fun();

* 1. var num = 1;

function fun () {

console.log(num);//值\_\_\_1\_\_\_\_

num = 2;

console.log(num);//值\_\_\_2\_\_\_\_

}

fun();

16、写出以下程序执行的结果

1. var a = 10;

a.pro = 10;

console.log(a.pro + a);

NAN number对象不可以定义私有属性 namber+非数字和字符的值就等于NaN

2)var s = ‘hello’;

s.pro = ‘world’;

console.log(s.pro + s); //undefinedhello s位字符串，字符串不可以自定义属性，所以s.pro为undefined 字符串做加运算会强制拼接位字符串

1. console.log(typeof fn);

function fn() {};

var fn;

//function 函数提升优先于变量提升

1. var f = true;

If(f === true) {

var a = 10;

}

function fn() {

var b = 20;

c = 30;

}

fn();

console.log(a);

//10

17、请看如下的代码，写出结果

var a = 5,b = 3;

function test() {

alert(b++);

var a = 4;

alert(--a);

alert(this.a);

}

1. tese(),三次alert()的值依次是什么？335 435 535
2. new test()，三次alert()的值依次是什么？ 33undefined 43undefined 53undefined //this更改了指向原来是指向window 用了new关键字后指向test test木有a属性所以为undefined

18、p最后显示什么颜色。怎么让p的颜色变成黑色，并简要说明css选择器优先级关系

<style>

#classA{color:yellow};

p.classB(color:red);

</style>

<body>  
 <p id=”classA” class=”classB”>123</p>

</body>

//p#classA{color：black}

19、关于正则表达式声明6位数字的邮编，一下代码正确的是(C)

1. var reg = /\d6/;
2. var reg = \d{6}\;
3. var reg = /\d{6}/;

D.var reg = new RegExp (“\d{6}”);

20、关于JavaScript里xml处理，一下说明正确的(A)

1. xml是种可扩展标记语言，格式更规范，是作为未来html的替代 //貌似XML是被替代的
2. Xml一般用于传输和存储数据，是对html的补充，两者的目的不同
3. 在JavaScript里解析和处理xml数据时，因为浏览器的不同，其做法也不同
4. 在IE浏览器里处理xml，首先需要创建ActiveXObject对象

21、请选择对javascript理解有误的(B)

1. javascript是网景公司开发的一种基于事件和驱动网页脚本语言
2. JScript是javascript的简称 //微软自己的浏览器才支持
3. FireFox和IE存在大量兼容性问题的主要原因在于他们对javascript的支持不同
4. AJAX技术一定要使用javascript技术

22、在Jquery中下面哪一个是用来追加到指定元素的末尾(B)

1. inserAfter()
2. Append()
3. appendTo()
4. After()

23、在javascript中定义变量var a=”35”,var b = “7”运算a % b 的结果为(C)

A.357

B.57

C.0

D.5

24、下面哪个属于javascript的字符型C

1. False
2. 你好
3. “123”
4. Null

25、下面哪个属于javascript的布尔值(C)

A.1.2

B.”true”

C.false

D.null

26、请选择结果为真的表达式(C)

1. null instanceof Object
2. Null === undefined;
3. null == undefined
4. NaN == NaN (false)

27、下列运算方式不属于逻辑运算的是(D)

1. !a

B.a&&b

C.a||b

**D**.a>b

28、声明一个对象，给它加上name属性和show方法显示其name值，以下代码中正确的是(D)

1. var obj = [name : “zhangsan” ,show : function(){alert(name);}];
2. Var obj = {name : “zhangsan”,show : “alert(this.name)”};
3. Var obj = {name : “zhangsan”,show : function () {alert(name);}};
4. Var obj = {name : “zhangsan”,show : function () {alert(this.name);}}

29、以下过于Array数组对象的说法不正确的是(C)

1. 对数组里数据的排序可以用sort函数，如果排序效果非预期，可以给sort函数加一个排序函数的参数
2. reverse用于对数组数据的倒序排列
3. 向数组的最后位置加一个新元素，可以用pop(删除数组中最后一个元素)方法 //push吧
4. unshift方法用于向数组删除一个元素

30、要将页面的状态显示”已经选中该文本”，下列JavaScript语句正确的是(A)

1. window.status = “已经选中该文本”
2. Document.status = “已经选中该文本”
3. Window.screen = “已经选中该文本”
4. Document.screen = “已经选中该文本”

31、点击页面的按钮，使之打开一个新窗口，加载一个页面，以下JavaScript代码中可执行的是(D)

1. <input type=”button” value=”new” onclick=”open(‘new.html’,’\_blank’)”>
2. <input type=”button” value=”new” onclick=”window.location=’new.html’;”>
3. <input type=”button” value=”new” onclick=”location.assign(‘new.html’);”>
4. <form target=”\_blank” action=”new.html”>

<input type=”submit” value=”new”>

</form>

32、下面的JavaScript语句中，实现检索当前页面中的表单元素中的所有文本框，并将它们全部清空(B)

1. for(var i = 0; i < form1.elements.length; i++) {

if(form1.elements[i].type ==”text”)

form1.elements[i].value = “”;

}

B.for (var i = 0; i < document.forms.length;i++) {

if(forms[0].elements[i].type == “text”)

form[0].elements[i].value = “”;

}

C.if(document.form.elements.type == “text”)

form.elements[i].value = “”;

D.for(var i = 0; i < document.forms.length;i++) {

for(var j = 0;j < document.forms[i].elements.length;j++) {

if(document.forms[i].elements[j].type == “text”)

document.forms[i].elements[j].value = “”;

}

}

33、在表单(form1)中有一个文本框元素(fname)，用于输入电话号码，格式如：010-82668155，要求前3位是010，紧接一个”-”，后面是8位数字。要求在提交表单时，根据上述条件验证该文本框中输入内容的有效性，下列语句中(A)能正确实现以上功能

1. var str = form1.fname.value;

If(str.substr(0,4)!=”010-”||str.substr(4).length!=8||isNaN(parseFloat(str.substr(4))))

Alert(“无效的电话号码！”);

B.var str = form1.fname.value;

If(str.substr(0,4)!=”010-”&&str.substr(4).length!=8&&isNaN(parseFloat(str.substr(4))))

Alert(“无效的电话号码！”);

C.var str = form1.fname.value;

If(str.substr(0,3)!=”010-”||str.substr(3).length!=8||isNaN(parseFloat(str.substr(3))))

alert(“无效的电话号码！”);

D.var str = form1.fname.value;

If(str.substr(0,4)!=”010-”&&str.substr(4).length!=8&&isNaN(parseFloat(str.substr(4))))

alert(“无效的电话号码！”);

34、关于正则表达式声明6位数字的邮编，一下代码正确的是(C)

A.var reg = /\d6/;

B.var reg = \d{6}\;

C.var reg = /\d{6}/;

D.var reg = new RegExp (“\d{6}”);

35、下面关于cookie的说明正确的是(D)

1. Cookie设置的过期时间为3600s是指60分钟过期
2. Cookie设置的过期时间为3600s是指只要在间隔60分钟内有动作时就不过期
3. Cookie保存在服务器端
4. Cookie保存在用户本地

36、使用js代码实现，将下面段落中含有的链接替换成可直接点击打开的链接

<p id=”text”>这个段落里有链接

比如：[http://www.abc.comm/和https://www.github.com/都是链接。](http://www.abc.comm/和https:/www.github.com/都是链接。)

可是他们显示在网页中是，链接不可点，还得复制粘贴到地址栏打开，好麻烦

</p>

37、写一个方法获取url？后面的参数，并将参数对象化。

function parseQueryString(url){

       var params = {};

       var arr = url.split("?");

       if (arr.length <= 1)

          return params;

       arr = arr[1].split("&");

       for(var i=0, l=arr.length; i<l; i++){

          var a = arr[i].split("=");

          params[a[0]] = a[1];

       }

       return params;

    }

    var url = "http://witmax.cn/index.php?key0=0&key1=1&key2=2";

    var ps = parseQueryString(url);

    console.log(ps["key1"]);

38、Node.js中，一段访问redis的代码如下

var redis = require(‘redis’);

var client = redis.createClient();

client.set(‘key’, ‘value’, function (err, data) {

if(err) {

console.error(err.message);

process.exit(1;)

}

cilent.get(‘key’,function (err,data) {

if(err) {

console.error(err.message);

return;

}

console.log(data);

process.exit(0);

})

});

请用Promise的异步调用方式重写

请用ES6 yield的异步调用方式重写

在经历了多个异步回调之后，如果拿到完整的堆信息(stack trace)?

39、用你认为合适的数据库产品，请设计数据结构，并完成一下方法(Server)；

1. 当出现一次网页浏览的时候，请实现函数click(url, ip)
2. 请实现查询函数pv(url)以及uv(url)

40、补充按钮事件的函数，确认用户是否退出当前页面，确认之后关闭窗口

<html>

<head>

<script type=”text/javasccript”>

Function closeWin() {

}

</script>

</head>

<body>

<input type=”button” value=”关闭窗口” onclick=”closeWin()” />

</body>

</html>

41、请用JavaScript实现，控制一个文本框只能输入正整数，如输入不符合条件则文本全部字体标红，要求写出完整的文本框HTML代码和JavaScript逻辑代码？

<!DOCTYPE html>

<**html**lang="en">

<**head**>

<**meta** charset="utf-8">

<**title**>只能输入正整数</**title**>

</**head**>

<**body**>

<**input** id="txt" type="text">

<**script**>

**var** txt= document.getElementById('txt');

**var** color = window.getComputedStyle(txt,'').color

txt.addEventListener('keyup',**function** () {

**var** reg = **new** RegExp("^[0-9]\*$");;

console.log(reg.test(**this**.value));

**if**(reg.test(**this**.value)){

**this**.style.color=color;

}**else**{

**this**.style.color='red';

}

});

</**script**>

</**body**>

</**html**>

42、请对以下代码进行优化

var wrap = document.getElementById(“wrap”);

for(var i = 0; i < 10; i++) {

var li = document.createElement(“li”);

var text = document.createTextNode(“hello” + i);

li.appendCChild(text);

wrap.appendChild(li);

}

43、请看下面的HTML，写出您的CSS使左边元素宽度为200px保持不变，右边元素随浏览器大小自适应

<div class=”outer”>

<div class=”left”></div>

<div class=”right”></div>

</div>

**非技术问题**

1、请概述一下你上家公司中项目的具体情况(工作所使用的技术,业务流程,周期,)？

我在上家公司做的的网站pc端页面的开发与维护。

主要技术：利用div+css布局，对用户注册页面开发，利用ajax技术与后台进行数据交互，使用Validation插件进行表单验证；使用bootstrap/zepto框架开发移动端页面，解决移动端设备的适配问题；侧边菜单栏的开发;利用css3完成轮播图动画

业务流程：根据需求分析，进行详细的总体设计，产生各栏目文件夹的结构图，根据美工的表现需要，设计静态网页和其它动态页面界面框架，程序员进行代码开发，做一些必要的测试，由项目组共同联调测试，发现bug，完善一些具体的细节。

周期：3个月

2、常用调试和优化工具？

Firebug+YSlow+其它Firefox扩展

浏览器自带工具，IE Developer Toolbar，Opera Dragonfly

Fiddler

HTTP Analyzer

HttpWatch

Web Developer

Web Accessibility Toolbar

3、什么叫代码部署？如何部署？

代码部署就是把开发好的网站代码放到应用服务器上对外提供服务、部署方式根据编程语言的不同而不同，但是大体的流程是一致的，生产环境主要是通过命令加配置文件的形式进行部署

4、新技术通过哪些渠道了解和学习？

优秀的博客，githup，http://www.daqianduan.com

5、对于前端这个岗位，兴趣的比例占多少？

70%

6、前端到底工作内容是什么？和UI有什么区别？

Web前端：主要讲UI提供的设计图，编码成静态html，实现所有特效;并负责所有交互的对接，对js要求较高

UI:主要对移动端和网页的设计

7、你当时进公司时是以什么身份进的，实习生吗？

是软件工程师，进公司之后直接上岗开发

8、工作中如果出现空档期的时候，你们都在做些什么？

答：空档期的话大家就都学习，学一些新的技术，也可以跟着js大神学习嘛，不过空档期的情况也很少，一般就几天。

9、平常在公司有做网页制作吗

答：如果实在忙的时候，我也帮制作师制作一些网页。

10、忙的时候，会帮网页制作做到什么程度，百分之多少？

答：一般忙的时候，我会先完成我自己的工作，像一些JS的特效和交互之后，再去帮网页制作，大概是10%。

11、你在你做过的哪个项目调试中，遇到了哪些比较深刻的部分，说一说。

你发现到解决这个问题用了多久？

答：通过你描述的问题难度，和你发现到解决问题的时间，看你js程度。

12、身为一位web前端工程师，你肯定知道现在最流行的前端技术有哪些吧？请例举3例？

浏览器兼容性，hack技术,Node.js

13、现有2个空水壶，容积分别为5升和6升，如何利用这两水壶取出3升水，假设水无限？

假设有A,B两只壶，A壶的容积为5升，B壶的容积为6升，

第一步：将B壶装满水，倒入A壶中，此时A壶满，B壶还剩一升

第二步：将A壶水到掉，将B壶水倒入,此时A壶为一升，B壶空

第三步：将B壶装满，倒入A壶中，此时A壶满，B壶还剩两升

第四步：重复第二步的操作，此时A壶剩两升，B壶空

第五步：重复第三步的操作，此时A壶满，B壶还剩三升

14、小明有100元去买汽水，汽水三元一瓶，正好小店有个促销活动，就是一个空瓶可以换1元钱，假设小明足够能喝，问他最多可以喝多少瓶汽水，还剩多少钱或空瓶？

49瓶汽水，还剩1元