前端面试题

刘于整理

前端开发知识点：

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 6、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

面试过程

* 代码编写笔试
* 问答试
* 过程一般遵循 STAR 面试法 。

STAR面试法，是企业招聘面试过程中可采用的技巧。其中，“STAR”是SITUATION（背景）、TASK（任务）、ACTION（行动）和RESULT（结果）四个英文单词的首字母组合。

问题举例编辑

问题：请讲出一件你通过学习尽快胜任新的工作任务的事。追问：

（1）这件事发生在什么时候？---------------------S

（2） 你要从事的工作任务是什么？--------------T

（3） 接到任务后你怎么办？-----------------------A

（4） 你用了多长时间获得完成该任务所必须的知识？------深层次了解

（5） 你在这个过程中遇见困难了吗？ -------------------顺便了解坚韧性

（6） 你最后完成任务的情况如何？--------------R

喜欢什么样的面试者

* 基础扎实

从多年的经验看，那些发展好的同学都具备扎实的基础知识

比如只懂 jQuery 不懂 JavaScript 是不行的哦

如果了解计算机基础会更好，因为我们将面临很多非前端技术的问题

* 主动思考

被动完成任务的同学在这里进步会很慢

你需要有自己的想法，而不是仅仅完成任务

* 爱学习

前端领域知识淘汰速度很快，所以最好能经常学习和接触新东西

* 有深度

遇到问题时多研究背后深层次的原因，而不是想办法先绕过去

比如追踪某个 Bug 一直了解它本质的原因

* 有视野

创新往往来自于不同学科的交集，如果你了解的领域越多，就越有可能有新想法

校招会更加关注基础知识，而社招会更加关注之前做过的项目情况。

HTML篇

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。

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。

3，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：

<br> <hr> <img> <input> <link> <meta>

鲜为人知的是：

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

4，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;

5，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 engineer或Rendering Engine)和JS引擎。

（1）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。

（2）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。

6，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）]

7，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;

语意化更好的内容元素，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样式。

当然也可以直接使用成熟的框架、比如html5shim;

<!--[if lt IE 9]>

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

<![endif]-->

\* 如何区分HTML5： DOCTYPE声明\新增的结构元素\功能元素

H5新特性

表单 画布 音视频 地理定位 媒体查询 css新特性 离线缓存 本地存储 拖拽

8，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。

9，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:

resourse/logo.png

FALLBACK:

/ /offline.html

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

10，浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢？

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

11，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），记会在浏览器和服务器间来回传递。

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

存储大小：

cookie数据大小不能超过4k。

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

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

12，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。

13，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>

14，如何实现浏览器内多个标签页之间的通信?

WebSocket、SharedWorker；

也可以调用localstorge、cookies等本地存储方式；

localstorge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localstorge值时会抛出 QuotaExceededError 的异常；

15，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

16，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：<strong>会重读，而<B>是展示强调内容。

i内容展示为斜体，em表示强调的文本；

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。

17，data-属性的作用是什么？

h5新增的属性 可以通过ele.dataset获取到标签上的data-\*的属性 返回一个对象

18，浏览器标准模式和怪异模式之间的区别是什么？

所谓的标准模式是指，浏览器按[W3C标准](https://www.baidu.com/s?wd=W3C%E6%A0%87%E5%87%86&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9m1Nbrj9hPvc1mym1PhPh0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHT1nWbkPj0sn1mznWT3nHfvPs)解析执行代码；怪异模式则是使用浏览器自己的方式解析执行代码，因为不同浏览器解析执行的方式不一样，所以我们称之为怪异模式。浏览器解析时到底使用标准模式还是怪异模式，与你网页中的DTD声明直接相关，DTD声明定义了标准文档的类型（标准模式解析）文档类型，会使浏览器使用相应的方式加载网页并显示，忽略DTD声明,将使网页进入怪异模式(quirks mode)。

所谓的标准模式是指，浏览器按[W3C标准](https://www.baidu.com/s?wd=W3C%E6%A0%87%E5%87%86&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9m1Nbrj9hPvc1mym1PhPh0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHT1nWbkPj0sn1mznWT3nHfvPs)解析执行代码；怪异模式则是使用浏览器自己的方式解析执行代码，因为不同浏览器解析执行的方式不一样，所以我们称之为怪异模式。浏览器解析时到底使用标准模式还是怪异模式，与你网页中的DTD声明直接相关，DTD声明定义了标准文档的类型（标准模式解析）文档类型，会使浏览器使用相应的方式加载网页并显示，忽略DTD声明,将使网页进入怪异模式(quirks mode)。

19，常见兼容性问题？

* + - 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游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 {}

20，你知道多少种Doctype文档类型？

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。

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，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。

21， HTML与XHTML——二者有什么区别？

区别：

1.所有的标记都必须要有一个相应的结束标记

2.所有标签的元素和属性的名字都必须使用小写

3.所有的XML标记都必须合理嵌套

4.所有的属性必须用引号""括起来

5.把所有<和&特殊符号用编码表示

6.给所有属性赋一个值

7.不要在注释内容中使“--”

8.图片必须有说明文字

22，<img>的title和alt有什么区别？

title是global attributes之一，用于为元素提供附加的advisory information。通常当鼠标滑动到元素上的时候显示。

alt是<img>的特有属性，是图片内容的等价描述，用于图片无法加载时显示、读屏器阅读图片。可提图片高可访问性，除了纯装饰图片外都必须设置有意义的值，搜索引擎会重点分析。

CSS篇

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，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 ;

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

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。

5，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

.div {

width:500px ; height:300px;//高度可以不设

margin: -150px 0 0 -250px;

position:relative; //相对定位

background-color:pink; //方便看效果

left:50%;

top:50%;

}

让绝对定位的div居中

position: absolute;

width: 1200px;

background: none;

margin: 0 auto;

top: 0;

left: 0;

bottom: 0;

right: 0;

6，display有哪些值？说明他们的作用。

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。

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

7，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。

8，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:

9，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;

}

10，一个满屏 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

11，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。

12，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; }

13，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

14，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。

创建规则：

根元素

浮动元素（float不是none）

绝对定位元素（position取值为absolute或fixed）

display取值为inline-block,table-cell, table-caption,flex, inline-flex之一的元素

overflow不是visible的元素

作用：

可以包含浮动元素

不被浮动元素覆盖

阻止父子元素的margin折叠

15，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

16，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

浮动元素脱离文档流，不占据空间。浮动元素碰到包含它的边框或者浮动元素的边框停留。

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；

16.1，浮动元素引起的问题和解决办法？

浮动元素引起的问题：

（1）父元素的高度无法被撑开，影响与父元素同级的元素

（2）与浮动元素同级的非浮动元素（内联元素）会跟随其后

（3）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

解决方法：

使用CSS中的clear:both;属性来清除元素的浮动可解决2、3问题，对于问题1，添加如下样式，给父元素添加clearfix样式：

.clearfix:after{content: ".";display: block;height: 0;clear: both;visibility: hidden;}

.clearfix{display: inline-block;} /\* for IE/Mac \*/

清除浮动的几种方法：

1，额外标签法，<div style="clear:both;"></div>（缺点：不过这个办法会增加额外的标签使HTML结构看起来不够简洁。）

2，使用after伪类

#parent:after{

content:"."; height:0; visibility:hidden; display:block; clear:both;

}

3,浮动外部元素

4,设置overflow为hidden或者auto

17，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。

18，使用 CSS 预处理器吗？喜欢那个？

SASS (SASS、LESS没有本质区别，只因为团队前端都是用的SASS)

19，CSS优化、提高性能的方法有哪些？

将样式表放到页面顶部

不使用CSS表达式

不使用@import

不使用IE的Filter

21，在网页中的应该使用奇数还是偶数的字体？为什么呢？

应该是用偶数字体

原因：

一，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

22，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。

何时应当时用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。

24，元素竖向的百分比设定是相对于容器的高度吗？

是的，元素的百分比设置一般是根据父级元素的宽高决定的，如果父级元素没有宽高百分比是不起作用的

25，全屏滚动的原理是什么？用到了CSS的那些属性？

全屏滚动的原理就是每次滚动滚轮，屏幕移动的距离就刚好是一个屏幕的距离，利用css的高度百分比实现这个效果，因为设定百分比的值是需要更具父元素设置的所以需要给html，body都设置为高度百分百，然后一层一层的设置，用js控制鼠标滚轮事件

26，什么是响应式设计？响应式设计的基本原理是什么？

响应式设计就是为了实现根据不同设备环境自动响应及调整网页布局的一种设计方案

基本原理就是利用css的媒体查询功能更具不同屏幕的大小，向下兼容设备、移动优先，达到响应的效果

28，::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。

双冒号是在当前规范中引入的，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和:after，单冒号和双冒号的写法::before和::after作用是一样的。

如果只需要兼容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。

29，如何修改chrome记住密码后自动填充表单的黄色背景 ？

chrome 表单自动填充后，input文本框的背景会变成黄色的，通过审查元素可以看到这是由于chrome会默认给自动填充的input表单加上input:-webkit-autofill私有属性，然后对其赋予以下样式：

input : -webkit-autofill {

background-color : #FAFFBD ; background-image : none ; color : #000 ;

}

在有些情况下，这个黄色的背景会影响到我们界面的效果，尤其是在我们给input文本框使用图片背景的时候，原来的圆角和边框都被覆盖了：

情景一：input文本框是纯色背景的

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；如：

input : -webkit-autofill {

-webkit-box-shadow : 0 0 0px 1000px white inset ; border : 1px solid #CCC !important ;

}

如果你有使用圆角等属性，或者发现输入框的长度高度不太对，可以对其进行调整，除了chrome默认定义的background-color，background-image，color不能用!important提升其优先级以外，其他的属性均可使用!important提升其优先级，如：

input : -webkit-autofill {

-webkit-box-shadow : 0 0 0px 1000px white inset ; border : 1px solid #CCC !important ;

height : 27px !important ; line-height : 27px !important ; border-radius : 0 4px 4px 0 ;

}

情景二：input文本框是使用图片背景的

这个比较麻烦，目前还没找到完美的解决方法，有两种选择：

1、如果你的图片背景不太复杂，只有一些简单的内阴影，那个人觉得完全可以使用上面介绍的方法用足够大的纯色内阴影去覆盖掉黄色背景，此时只不过是没有了原来的内阴影效果罢了。

2、如果你实在想留住原来的内阴影效果，那就只能牺牲chrome自动填充表单的功能，使用 js 去实现，例如：

$ ( function () {

if ( navigator . userAgent . toLowerCase (). indexOf ( "chrome" ) >= 0 ) {

$ ( window ). load ( function (){

$ ( 'ul input:not(input[type=submit])' ). each ( function (){

var outHtml = this . outerHTML ;

$ ( this ). append ( outHtml );

});

});

}

});

遍历的对象可能要根据你的需求去调整。如果你不想使用js，好吧，在form标签上直接关闭了表单的自动填充功能：autocomplete="off"。

30，你对line-height是如何理解的？

line－height是用来设置字体的行高，可以理解为设置文字高度的占位大小，单位可以是数字，或者像素

数字：更具文字的大小来决定 1.5相当于文字大小的1.5倍

像素：就是通常的px大小了

31，怎么让Chrome支持小于12px 的文字？

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

33，position:fixed;在android下无效怎么处理？

可以使用css的变形功能对文字缩放

34，如果需要手动写动画，你认为最小时间间隔是多久，为什么？

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

35，display:inline-block 什么时候会显示间隙？

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

37，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。

box-sizing方案

外层box-sizing: border-box; 同时设置padding: 100px 0 0；

内层100像素高的元素向上移动100像素，或使用absolute定位防止占据空间；

另一个元素直接height: 100%;

absolute positioning

外层position: relative；

百分百自适应元素直接position: absolute; top: 100px; bottom: 0; left: 0

js计算

38，png、jpg、gif 这些图片格式解释一下，分别什么时候用？

GIF:

8位像素，256色 无损压缩 支持简单动画 支持boolean透明

适合简单动画

JPEG：

颜色限于256 有损压缩 可控制压缩质量 不支持透明 适合照片

PNG：

有PNG8和truecolor PNG PNG8类似GIF颜色上限为256，文件小，支持alpha透明度，无动画

适合图标、背景、按钮

39，什么是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。

40，style标签写在body后与body前有什么区别？

可以的，但是CSS又会重新渲染了一次页面，占用一定的时间，如果网页篇幅少的话感觉不出来， 如果篇幅多（包含有大量HTML）会有一定的影响，所以大页面中不建议将style写进body里。

41，什么是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。

42，什么是CSS reset？

重置浏览器标签的样式表

在HTML标签在浏览器里有默认的样式，例如 p 标签有上下边距，strong标签有字体加粗样式，em标签有字体倾斜样式。不同浏览器的默认样式之间也会有差别，例如ul默认带有缩进的样式，在IE下，它的缩进是通过margin实现的，而Firefox下，它的缩进是由padding实现的。在切换页面的时候，浏览器的默认样式往往会给我们带来麻烦，影响开发效率。所以解决的方法就是一开始就将浏览器的默认样式全部去掉，更准确说就是通过重新定义标签样式。“覆盖”浏览器的CSS默认属性。最最简单的说法就是把浏览器提供的默认样式覆盖掉！这就是CSS reset。

44，css sprite是什么,有什么优缺点。

概念：将多个小图片拼接到一个图片中。通过background-position和元素尺寸调节需要显示的背景图案

优点：

减少HTTP请求数，极大地提高页面加载速度

增加图片信息重复度，提高压缩比，减少图片大小

更换风格方便，只需在一张或几张图片上修改颜色或样式即可实现

缺点：

图片合并麻烦

维护麻烦，修改一个图片可能需要从新布局整个图片，样式

45，你最喜欢的图片替换方法是什么，你如何选择使用。，

46，你用过栅格系统吗？如果使用过，你最喜欢哪种？

大多数ui框架中栅格系统，我感觉栅格系统都差不多，如果非要说一个的话，我还是比较喜欢使用bootstrap

47，渐进增强和优雅降级

渐进增强 ：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 ：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。

48，什么是 FOUC（无样式内容闪烁）？你如何来避免 FOUC？

FOUC - Flash Of Unstyled Content 文档样式闪烁

<style type="text/css" media="all">@import "../fouc.css";</style>

而引用CSS文件的@import就是造成这个问题的罪魁祸首。IE会先加载整个HTML文档的DOM，然后再去导入外部的CSS文件，因此，在页面DOM加载完成到CSS导入完成中间会有一段时间页面上的内容是没有样式的，这段时间的长短跟网速，电脑速度都有关系。

解决方法简单的出奇，只要在<head>之间加入一个<link>或者<script>元素就可以了。

49，CSS选择器有哪些

<div id="demo">  
 <div class="inner">  
 <p><a href="#">链接</a></p>  
 <p><input type="checkbox" name="" id="" /></p>  
 </div>  
</div>

class 选择器 .inner{ width:100px;}  
id 选择器 #demo{ width:100px;}  
标签 选择器 p{ width:100px;}  
后代选择器 p a{ color: #f60;}  
伪类选择器 a[:hover](https://www.baidu.com/s?wd=:hover&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dWnjbzuhwhuyc4rjPBmH6v0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjn3nHTLrjbs){ color: #f00;}  
伪元素选择器 a[:after](https://www.baidu.com/s?wd=:after&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dWnjbzuhwhuyc4rjPBmH6v0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjn3nHTLrjbs){ clear: both;}  
属性选择器 input[type=checkbox]{ [margin-left](https://www.baidu.com/s?wd=margin-left&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dWnjbzuhwhuyc4rjPBmH6v0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjn3nHTLrjbs): 10px;}  
由于IE，一般用前四种

50，display: none;与visibility: hidden;的区别

联系：它们都能让元素不可见

区别：一个隐藏占位，一个隐藏不占位

display:none;会让元素完全从渲染树中消失，渲染的时候不占据任何空间；visibility: hidden;不会让元素从渲染树消失，渲染师元素继续占据空间，只是内容不可见

display: none;是非继承属性，子孙节点消失由于元素从渲染树消失造成，通过修改子孙节点属性无法显示；visibility: hidden;是继承属性，子孙节点消失由于继承了hidden，通过设置visibility: visible;可以让子孙节点显式

修改常规流中元素的display通常会造成文档重排。修改visibility属性只会造成本元素的重绘。

读屏器不会读取display: none;元素内容；会读取visibility: hidden;元素内容

51，rem和em

rem和em都是相对设置大小的单位，而且都是相对于字体大小

rem想对于跟节点html的文字大小

em想对于父级元素的字体大小设置

52，css3中的动画

css中动画功能分为transition和animation两种，这个两种方法都可以通过改变css中的属性值来产生动画效果

transition：只能实现两个简单值之间的过渡

animation：通过引用keyframes关键帧来实现复杂动画

53，css3动画的性能优化你知道什么？

JavaScript篇

1，介绍js的基本数据类型。

Undefined、Null、Boolean、Number、String

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，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。

4，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()}

5，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

6，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;

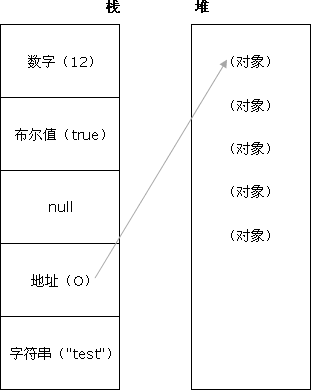
}

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



7，JavaScript继承的几种实现方式？

四种，构造继承 原型继承 实例继承 拷贝继承

8，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();

5、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();

9，Javascript作用链域?

作用域链的作用是保证执行环境里有权访问的变量和函数是有序的，作用域链的变量只能向上访问，变量访问到window对象即被终止，作用域链向下访问变量是不被允许的。

11，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；

this就是指的是当前对象

10，什么是window对象? 什么是document对象?

它是一个顶层对象,而不是另一个对象的属性，即浏览器的窗口。

window对象：它是一个顶层对象，不是另一个对象的属性，及浏览器窗口

Document对象：是window对象的一个属性，是显示窗口内的一个文档

[document 对象]

　　该对象是window和frames对象的一个属性,是显示于窗口或框架内的

11，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

undefined表示"缺少值"，就是此处应该有一个值，但是还没有定义。典型用法是：

（1）变量被声明了，但没有赋值时，就等于undefined。

（2) 调用函数时，应该提供的参数没有提供，该参数等于undefined。

（3）对象没有赋值的属性，该属性的值为undefined。

（4）函数没有返回值时，默认返回undefined。

null表示"没有对象"，即该处不应该有值。典型用法是：

（1） 作为函数的参数，表示该函数的参数不是对象。

（2） 作为对象原型链的终点。

12，写一个通用的事件侦听器函数。

// event(事件)工具集，来源：github.com/markyun

markyun.Event = {

// 页面加载完成后

readyEvent : function(fn) {

if (fn==null) {

fn=document;

}

var oldonload = window.onload;

if (typeof window.onload != 'function') {

window.onload = fn;

} else {

window.onload = function() {

oldonload();

fn();

};

}

},

// 视能力分别使用dom0||dom2||IE方式 来绑定事件

// 参数： 操作的元素,事件名称 ,事件处理程序

addEvent : function(element, type, handler) {

if (element.addEventListener) {

//事件类型、需要执行的函数、是否捕捉

element.addEventListener(type, handler, false);

} else if (element.attachEvent) {

element.attachEvent('on' + type, function() {

handler.call(element);

});

} else {

element['on' + type] = handler;

}

},

// 移除事件

removeEvent : function(element, type, handler) {

if (element.removeEventListener) {

element.removeEventListener(type, handler, false);

} else if (element.datachEvent) {

element.detachEvent('on' + type, handler);

} else {

element['on' + type] = null;

}

},

// 阻止事件 (主要是事件冒泡，因为IE不支持事件捕获)

stopPropagation : function(ev) {

if (ev.stopPropagation) {

ev.stopPropagation();

} else {

ev.cancelBubble = true;

}

},

// 取消事件的默认行为

preventDefault : function(event) {

if (event.preventDefault) {

event.preventDefault();

} else {

event.returnValue = false;

}

},

// 获取事件目标

getTarget : function(event) {

return event.target || event.srcElement;

},

// 获取event对象的引用，取到事件的所有信息，确保随时能使用event；

getEvent : function(e) {

var ev = e || window.event;

if (!ev) {

var c = this.getEvent.caller;

while (c) {

ev = c.arguments[0];

if (ev && Event == ev.constructor) {

break;

}

c = c.caller;

}

}

return ev;

}

};

13，["1", "2", "3"].map(parseInt) 答案是多少？

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

其中 radix 表示解析时用的基数。

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

14，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;）

15，什么是闭包（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链域，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

16，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。

17，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');

}

18，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);

19，用原生JavaScript的实现过什么功能吗？

20，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(proName)

其中参数object是必选项。一个对象的实例。

proName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。

21，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);

23，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

defer 属性,该属性用来通知浏览器，这段脚本代码将不会产生任何文档内容。例如 JavaScript代码中的document.write()方法将不会骑作用，浏览器遇到这样的代码将会忽略，并继续执行后面的代码。属性只能是 defer，与属性名相同。在HTML语法格式下，也允许不定义属性值，仅仅使用属性名。加上 defer 等于在页面完全在入后再执行.

<scriptsrc="file.js" defer></script>

async 属性,不能保证脚本会按顺序执行。它们将在onload 事件之前完成。async和defer一样，都不会阻塞其他资源下载，所以不会影响页面的加载，但在async的情况下，js文档一旦下载完毕就会立刻执行，所以很有可能不是按照原本的顺序来执行

<scriptsrc="file.js" async></script>

动态创建DOM方式 通过js创建DOM节点 动态添加

28，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

　　　　 };

　　 })();

33，异步加载JS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

34，documen.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

35，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

getElementsByTagName() //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

36，.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);

add.apply(sub,[3,1]);

功能一样，但是传递参赛的方式不一样

37，数组和对象有哪些原生方法，列举一下？

push() pop() join() indexOf() …

39，JavaScript中的作用域？

在js中每一个方法都试一个作用域，最大的作用域就是window对象，

作用域中的变量和方法只能在当前作用域和当前作用域下的作用域使用

外面的作用域不能访问里面作用域的变量和方法

js中查找变量的方式是，先在当前作用域查找，如果没有的话，就去上一级作用域中查找

40，如何编写高性能的Javascript？

41，那些操作会造成内存泄漏？

使用闭包的时候如果在闭包变量中保存了大量的dom结构而且不去使用长期存在的时候

内存泄漏指任何对象在您不再拥有或需要它之后仍然存在。

垃圾回收器定期扫描对象，并计算引用了每个对象的其他对象的数量。如果一个对象的引用数量为 0（没有其他对象引用过该对象），或对该对象的惟一引用是循环的，那么该对象的内存即可回收。

setTimeout 的第一个参数使用字符串而非函数的话，会引发内存泄漏。

闭包、控制台日志、循环（在两个对象彼此引用且彼此保留时，就会产生一个循环）

42，页面重构怎么操作？

网站重构：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。

也就是说是在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的UI。

对于传统的网站来说重构通常是：

表格(table)布局改为DIV+CSS

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)

对于移动平台的优化

针对于SEO进行优化

深层次的网站重构应该考虑的方面

减少代码间的耦合

让代码保持弹性

严格按规范编写代码

设计可扩展的API

代替旧有的框架、语言(如VB)

增强用户体验

通常来说对于速度的优化也包含在重构中

压缩JS、CSS、image等前端资源(通常是由服务器来解决)

程序的性能优化(如数据读写)

采用CDN来加速资源加载

对于JS DOM的优化

HTTP服务器的文件缓存

43，列举IE与其他浏览器不一样的特性？

1、事件不同之处：

触发事件的元素被认为是目标（target）。而在 IE 中，目标包含在 event 对象的 srcElement 属性；

获取字符代码、如果按键代表一个字符（shift、ctrl、alt除外），IE 的 keyCode 会返回字符代码（Unicode），DOM 中按键的代码和字符是分离的，要获取字符代码，需要使用 charCode 属性；

阻止某个事件的默认行为，IE 中阻止某个事件的默认行为，必须将 returnValue 属性设置为 false，Mozilla 中，需要调用 preventDefault() 方法；

停止事件冒泡，IE 中阻止事件进一步冒泡，需要设置 cancelBubble 为 true，Mozzilla 中，需要调用 stopPropagation()；

44，什么叫优雅降级和渐进增强？

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

如：border-shadow

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

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

45，是否了解公钥加密和私钥加密。

一般情况下是指私钥用于对数据进行签名，公钥用于对签名进行验证;

HTTP网站在浏览器端用公钥加密敏感数据，然后在服务器端再用私钥解密。

46，WEB应用从服务器主动推送Data到客户端有那些方式？

html5提供的Websocket

不可见的iframe

WebSocket通过Flash

XHR长时间连接

XHR Multipart Streaming

<script>标签的长时间连接(可跨域)

47，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，

因此客户端和服务器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。

48，你有用过哪些前端性能优化的方法？

（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，显示比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。

49，解释下事件代理。

事件目标 target

50，解释下 JavaScript 中 this 是如何工作的。

作者：甘洪翔  
链接：https://www.zhihu.com/question/19624483/answer/24422902  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

 方法调用模式  
当函数被保存为对象的一个属性时，成该函数为该对象的方法。函数中this的值为该对象。

 var foo = {

name: 'fooname',

getName: function (){

return this.name

}

}

foo.getName(); // this => foo

  函数调用模式  
当函数并不是对象的属性。函数中this的值为全局对象  
note：某个方法中的内部函数中的this的值也是全局对象，而非外部函数的this

 function foo(){

this.name = 'fooname';

}

foo(); // this => window

  构造器调用模式  
即使用new调用的函数，则其中this将会被绑定到那个新构造的对象。

 function Foo(){

this.name = 'fooname';

}

var foo = new Foo(); // this => foo

  使用apply或call调用模式  
该模式调用时，函数中this被绑定到apply或call方法调用时接受的第一个参数。

 function getName(name){

this.name = name;

}

var foo = {};

getName.call(foo, name); // this =>foo



52，什么是哈希表？

53，请举出一个匿名函数的典型用例？

54，你何时优化自己的代码？

一般都是功能完成后，不着急做其他功能的时候，会选择对以前实习的功能进行代码优化

55，== 和 === 有什么不同？

＝＝是派生自＝＝＝，＝＝回自动转化数据类型，＝＝＝不会

56，你如何从浏览器的 URL 中获取查询字符串参数。

Location.search

59，eval是做什么的？

**eval()的作用**

把字符串参数解析成JS代码并运行，并返回执行的结果

eval("2+3");//执行加运算，并返回运算值。

eval("varage=10");//声明一个age变量

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

IE8及以下不兼容

functiona(){

**if**(window.execScript){ // 支持IE8及以下的版本

window.execScript("var x=1");

}

**else**{ //常用的浏览器都支持

 window.eval("var x=1");

}

console.log(x);

}

a();

console.log(x);

由JSON字符串转换为JSON对象的时候可以用eval

Var json="{name:'Mr.CAO',age:30}";

Var jsonObj=eval("("+json+")");

60，Javascript垃圾回收方法

标记清除（mark and sweep）

这是JavaScript最常见的垃圾回收方式，当变量进入执行环境的时候，比如函数中声明一个变量，垃圾回收器将其标记为“进入环境”，当变量离开环境的时候（函数执行结束）将其标记为“离开环境”。

垃圾回收器会在运行的时候给存储在内存中的所有变量加上标记，然后去掉环境中的变量以及被环境中变量所引用的变量（闭包），在这些完成之后仍存在标记的就是要删除的变量了

引用计数(reference counting)

在低版本IE中经常会出现内存泄露，很多时候就是因为其采用引用计数方式进行垃圾回收。引用计数的策略是跟踪记录每个值被使用的次数，当声明了一个 变量并将一个引用类型赋值给该变量的时候这个值的引用次数就加1，如果该变量的值变成了另外一个，则这个值得引用次数减1，当这个值的引用次数变为0的时 候，说明没有变量在使用，这个值没法被访问了，因此可以将其占用的空间回收，这样垃圾回收器会在运行的时候清理掉引用次数为0的值占用的空间。

在IE中虽然JavaScript对象通过标记清除的方式进行垃圾回收，但BOM与DOM对象却是通过引用计数回收垃圾的， 也就是说只要涉及BOM及DOM就会出现循环引用问题。

61，用过哪些设计模式？

工厂模式：

主要好处就是可以消除对象间的耦合，通过使用工程方法而不是new关键字。将所有实例化的代码集中在一个位置防止代码重复。

工厂模式解决了重复实例化的问题 ，但还有一个问题,那就是识别问题，因为根本无法 搞清楚他们到底是哪个对象的实例。

function createObject(name,age,profession){//集中实例化的函数var obj = new Object();

obj.name = name;

obj.age = age;

obj.profession = profession;

obj.move = function () {

return this.name + ' at ' + this.age + ' engaged in ' + this.profession;

};

return obj;

}

var test1 = createObject('trigkit4',22,'programmer');//第一个实例var test2 = createObject('mike',25,'engineer');//第二个实例

构造函数模式

使用构造函数的方法 ，即解决了重复实例化的问题 ，又解决了对象识别的问题，该模式与工厂模式的不同之处在于：

1.构造函数方法没有显示的创建对象 (new Object());

2.直接将属性和方法赋值给 this 对象;

3.没有 renturn 语句。

62，请你谈谈Cookie的弊端

cookie虽然在持久保存客户端数据提供了方便，分担了服务器存储的负担，但还是有很多局限性的。

第一：每个特定的域名下最多生成20个cookie

1.IE6或更低版本最多20个cookie

2.IE7和之后的版本最后可以有50个cookie。

3.Firefox最多50个cookie

4.chrome和Safari没有做硬性限制

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

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

IE 提供了一种存储可以持久化用户数据，叫做userdata，从IE5.0就开始支持。每个数据最多128K，每个域名下最多1M。这个持久化数据放在缓存中，如果缓存没有清理，那么会一直存在。

优点：极高的扩展性和可用性

1.通过良好的编程，控制保存在cookie中的session对象的大小。

2.通过加密和安全传输技术（SSL），减少cookie被破解的可能性。

3.只在cookie中存放不敏感数据，即使被盗也不会有重大损失。

4.控制cookie的生命期，使之不会永远有效。偷盗者很可能拿到一个过期的cookie。

缺点：

1.`Cookie`数量和长度的限制。每个domain最多只能有20条cookie，每个cookie长度不能超过4KB，否则会被截掉.

2.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。即使加密也与事无补，因为拦截者并不需要知道cookie的意义，他只要原样转发cookie就可以达到目的了。

3.有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

63，浏览器本地存储

在较高版本的浏览器中，js提供了sessionStorage和globalStorage。在HTML5中提供了localStorage来取代globalStorage。

html5中的Web Storage包括了两种存储方式：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。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储

而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。

64，cookie 和session 的区别：

1、cookie数据存放在客户的浏览器上，session数据放在服务器上。

2、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗

考虑到安全应当使用session。

3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能

考虑到减轻服务器性能方面，应当使用COOKIE。

4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

5、所以个人建议：

将登陆信息等重要信息存放为SESSION

其他信息如果需要保留，可以放在COOKIE中

65，说说严格模式的限制

严格模式主要有以下限制：

变量必须声明后再使用

函数的参数不能有同名属性，否则报错

不能使用with语句

不能对只读属性赋值，否则报错

不能使用前缀0表示八进制数，否则报错

不能删除不可删除的属性，否则报错

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

eval不会在它的外层作用域引入变量

eval和arguments不能被重新赋值

arguments不会自动反映函数参数的变化

不能使用

不能使用arguments.caller

禁止this指向全局对象

不能使用fn.caller和fn.arguments获取函数调用的堆栈

增加了保留字（比如protected、static和interface）

设立"严格模式"的目的，主要有以下几个：

消除Javascript语法的一些不合理、不严谨之处，减少一些怪异行为;

消除代码运行的一些不安全之处，保证代码运行的安全；

提高编译器效率，增加运行速度；

为未来新版本的Javascript做好铺垫。

注：经过测试IE6,7,8,9均不支持严格模式。

66，document.write()的用法

document.write()方法可以用在两个方面：页面载入过程中用实时脚本创建页面内容，以及用延时脚本创建本窗口或新窗口的内容。

document.write只能重绘整个页面。innerHTML可以重绘页面的一部分

67，编写一个方法 求一个字符串的字节长度

假设：一个英文字符占用一个字节，一个中文字符占用两个字节

function GetBytes(str){

var len = str.length;

var bytes = len;

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

if (str.charCodeAt(i) > 255) bytes++;

}

return bytes;

}

alert(GetBytes(“你好,as"));

68，说说你对MVC和MVVM的理解

MVC

View 传送指令到 Controller

Controller 完成业务逻辑后，要求 Model 改变状态

Model 将新的数据发送到 View，用户得到反馈

所有通信都是单向的。

Angular它采用双向绑定（data-binding）：View的变动，自动反映在 ViewModel，反之亦然。

组成部分Model、View、ViewModel

View：UI界面

ViewModel：它是View的抽象，负责View与Model之间信息转换，将View的Command传送到Model；

Model：数据访问层

69，说说网络分层里七层模型是哪七层

应用层：应用层、表示层、会话层（从上往下）（HTTP、FTP、SMTP、DNS）

传输层（TCP和UDP）

网络层（IP）

物理和数据链路层（以太网）

每一层的作用如下：

物理层：通过媒介传输比特,确定机械及电气规范（比特Bit）

数据链路层：将比特组装成帧和点到点的传递（帧Frame）

网络层：负责数据包从源到宿的传递和网际互连（包PackeT）

传输层：提供端到端的可靠报文传递和错误恢复（段Segment）

会话层：建立、管理和终止会话（会话协议数据单元SPDU）

表示层：对数据进行翻译、加密和压缩（表示协议数据单元PPDU）

应用层：允许访问OSI环境的手段（应用协议数据单元APDU）

70，什么样的前端代码是好的

高复用低耦合，这样文件小，好维护，而且好扩展。

71，attribute和property的区别是什么？

attribute是dom元素在文档中作为html标签拥有的属性；

property就是dom元素在js中作为对象拥有的属性。

所以：

对于html的标准属性来说，attribute和property是同步的，是会自动更新的，

但是对于自定义的属性来说，他们是不同步的，

72，说说你对AMD和Commonjs的理解

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

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

73，offsetWidth/offsetHeight,clientWidth/clientHeight与scrollWidth/scrollHeight的区别，

offsetWidth/offsetHeight返回值包含content + padding + border，效果与e.getBoundingClientRect()相同

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

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

74，focus/blur与focusin/focusout的区别与联系

focus/blur不冒泡，focusin/focusout冒泡

focus/blur兼容性好，focusin/focusout在除FireFox外的浏览器下都保持良好兼容性，如需使用事件托管，可考虑在FireFox下使用事件捕获elem.addEventListener('focus', handler, true)

可获得焦点的元素：

window

链接被点击或键盘操作

表单空间被点击或键盘操作

设置tabindex属性的元素被点击或键盘操作

75，mouseover/mouseout与mouseenter/mouseleave的区别与联系

mouseover/mouseout是标准事件，所有浏览器都支持；mouseenter/mouseleave是IE5.5引入的特有事件后来被DOM3标准采纳，现代标准浏览器也支持

mouseover/mouseout是冒泡事件；mouseenter/mouseleave不冒泡。需要为多个元素监听鼠标移入/出事件时，推荐mouseover/mouseout托管，提高性能

标准事件模型中event.target表示发生移入/出的元素,vent.relatedTarget对应移出/如元素；在老IE中 event.srcElement表示发生移入/出的元素，event.toElement表示移出的目标元素，event.fromElement表示移入时的来源元素

76，应用程序存储和离线web应用

HTML5新增应用程序缓存，允许web应用将应用程序自身保存到用户浏览器中，用户离线状态也能访问。

1.为html元素设置manifest属性:<html manifest="myapp.appcache">，其中后缀名只是一个约定，真正识别方式是通过text/cache-manifest作为MIME类型。所以需要配置服务器保证设置正确

2.manifest文件首行为CACHE MANIFEST，其余就是要缓存的URL列表，每个一行，相对路径都相对于manifest文件的url。注释以#开头

3.url分为三种类型：CACHE:为默认类型。NETWORK：表示资源从不缓存。 FALLBACK:每行包含两个url，第二个URL是指需要加载和存储在缓存中的资源， 第一个URL是一个前缀。任何匹配该前缀的URL都不会缓存，如果从网络中载入这样的URL失败的话，就会用第二个URL指定的缓存资源来替代。以下是一个文件例子：

CACHE MANIFEST

CACHE:

myapp.html

myapp.css

myapp.js

FALLBACK:

videos/ offline\_help.html

NETWORK:

cgi/

77，请用代码写出(今天是星期x)其中x表示当天是星期几,如果当天是星期一,输出应该是”今天是星期一”

var days = ['日','一','二','三','四','五','六'];

var date = new Date();

console.log('今天是星期' + days[date.getDay()]);

78，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,请说明为什么,并修改循环内的代码使其输出正确结果

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

setTimeout(function () {

console.log(i + ' ');

}, 100);

}

不能输出正确结果，因为循环中setTimeout接受的参数函数通过闭包访问变量i。javascript运行环境为单线程，setTimeout注册的函数需要等待线程空闲才能执行，此时for循环已经结束，i值为5.五个定时输出都是5 修改方法：将setTimeout放在函数立即调用表达式中，将i值作为参数传递给包裹函数，创建新闭包

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

(function (i) {

setTimeout(function () {

console.log(i + ' ');

}, 100);

}(i));

}

AJAX篇

1，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

ajax是一种创建交互式网页的计算

2，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，如何解决跨域问题?

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

参考：<http://qiutc.me/post/cross-domain-collections.html>

4，http状态码有那些？分别代表是什么意思？

简单版

[

100 Continue 继续，一般在发送post请求时，已发送了http header之后服务端将返回此信息，表示确认，之后发送具体参数信息

200 OK 正常返回信息

201 Created 请求成功并且服务器创建了新的资源

202 Accepted 服务器已接受请求，但尚未处理

301 Moved Permanently 请求的网页已永久移动到新位置。

302 Found 临时性重定向。

303 See Other 临时性重定向，且总是使用 GET 请求新的 URI。

304 Not Modified 自从上次请求后，请求的网页未修改过。

400 Bad Request 服务器无法理解请求的格式，客户端不应当尝试再次使用相同的内容发起请求。

401 Unauthorized 请求未授权。

403 Forbidden 禁止访问。

404 Not Found 找不到如何与 URI 相匹配的资源。

500 Internal Server Error 最常见的服务器端错误。

503 Service Unavailable 服务器端暂时无法处理请求（可能是过载或维护）。

]

5，一个页面从输入 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）

注：这题胜在区分度高，知识点覆盖广，再不懂的人，也能答出几句，

而高手可以根据自己擅长的领域自由发挥，从URL规范、HTTP协议、DNS、CDN、数据库查询、到浏览器流式解析、CSS规则构建、layout、paint、onload/domready、JS执行、JS API绑定等等；

详细版：

1、浏览器会开启一个线程来处理这个请求，对 URL 分析判断如果是 http 协议就按照 Web 方式来处理;

2、调用浏览器内核中的对应方法，比如 WebView 中的 loadUrl 方法;

3、通过DNS解析获取网址的IP地址，设置 UA 等信息发出第二个GET请求;

4、进行HTTP协议会话，客户端发送报头(请求报头);

5、进入到web服务器上的 Web Server，如 Apache、Tomcat、Node.JS 等服务器;

6、进入部署好的后端应用，如 PHP、Java、JavaScript、Python 等，找到对应的请求处理;

7、处理结束回馈报头，此处如果浏览器访问过，缓存上有对应资源，会与服务器最后修改时间对比，一致则返回304;

8、浏览器开始下载html文档(响应报头，状态码200)，同时使用缓存;

9、文档树建立，根据标记请求所需指定MIME类型的文件（比如css、js）,同时设置了cookie;

10、页面开始渲染DOM，JS根据DOM API操作DOM,执行事件绑定等，页面显示完成。

简洁版：

浏览器根据请求的URL交给DNS域名解析，找到真实IP，向服务器发起请求；

服务器交给后台处理完成后返回数据，浏览器接收文件（HTML、JS、CSS、图象等）；

浏览器对加载到的资源（HTML、JS、CSS等）进行语法解析，建立相应的内部数据结构（如HTML的DOM）；

载入解析到的资源文件，渲染页面，完成。

7，从打开app到刷新出内容，整个过程中都发生了什么，如果感觉慢，怎么定位问题，怎么解决?

8，请解释 JSONP 的工作原理，以及它为什么不是真正的 AJAX。

jsonp的工作原理是，动态的创建了一个全局方法，并且动态生成script标签请求数据，在请求回来的数据中是执行请求是动态生成的js方法，并且传递参数是请求的数据，生成了假象的ajax

9，请解释一下 JavaScript 的同源策略。

同源策略是客户端脚本（尤其是Javascript）的重要的安全度量标准。它最早出自Netscape Navigator2.0，其目的是防止某个文档或脚本从多个不同源装载。

它的精髓很简单：它认为自任何站点装载的信赖内容是不安全的。当被浏览器半信半疑的脚本运行在沙箱时，它们应该只被允许访问来自同一站点的资源，而不是那些来自其它站点可能怀有恶意的资源。

这里的同源指的是：同协议，同域名和同端口

10，为什么要有同源限制？

我们举例说明：比如一个黑客程序，他利用IFrame把真正的银行登录页面嵌到他的页面上，当你使用真实的用户名，密码登录时，他的页面就可以通过Javascript读取到你的表单中input中的内容，这样用户名，密码就轻松到手了。

11，创建ajax过程

(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.

var xmlHttp = new XMLHttpRequest();

xmlHttp.open('GET','demo.php','true');

xmlHttp.send()

xmlHttp.onreadystatechange = function(){

if(xmlHttp.readyState === 4 & xmlHttp.status === 200){

}

}

11，常见web安全及防护原理

sql注入原理

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

总的来说有以下几点：

1.永远不要信任用户的输入，要对用户的输入进行校验，可以通过正则表达式，或限制长度，对单引号和双"-"进行转换等。

2.永远不要使用动态拼装SQL，可以使用参数化的SQL或者直接使用存储过程进行数据查询存取。

3.永远不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的数据库连接。

4.不要把机密信息明文存放，请加密或者hash掉密码和敏感的信息。

XSS原理及防范

Xss(cross-site scripting)攻击指的是攻击者往Web页面里插入恶意 html标签或者javascript代码。比如：攻击者在论坛中放一个

看似安全的链接，骗取用户点击后，窃取cookie中的用户私密信息；或者攻击者在论坛中加一个恶意表单

当用户提交表单的时候，却把信息传送到攻击者的服务器中，而不是用户原本以为的信任站点。

XSS防范方法

首先代码里对用户输入的地方和变量都需要仔细检查长度和对”<”,”>”,”;”,”’”等字符做过滤；其次任何内容写到页面之前都必须加以encode，避免不小心把html tag 弄出来。这一个层面做好，至少可以堵住超过一半的XSS 攻击。

首先，避免直接在cookie 中泄露用户隐私，例如email、密码等等。

其次，通过使cookie 和系统ip 绑定来降低cookie 泄露后的危险。这样攻击者得到的cookie 没有实际价值，不可能拿来重放。

如果网站不需要再浏览器端对cookie 进行操作，可以在Set-Cookie 末尾加上HttpOnly 来防止javascript 代码直接获取cookie 。

尽量采用POST 而非GET 提交表单

XSS与CSRF有什么区别吗？

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

要完成一次CSRF攻击，受害者必须依次完成两个步骤：

登录受信任网站A，并在本地生成Cookie。

在不登出A的情况下，访问危险网站B。

CSRF的防御

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

通过验证码的方法

12，HTTP和HTTPS

HTTP协议通常承载于TCP协议之上，在HTTP和TCP之间添加一个安全协议层（SSL或TSL），这个时候，就成了我们常说的HTTPS。

默认HTTP的端口号为80，HTTPS的端口号为443。

13，为什么HTTPS安全

因为网络请求需要中间有很多的服务器路由器的转发。中间的节点都可能篡改信息，而如果使用HTTPS，密钥在你和终点站才有。https之所以比http安全，是因为他利用ssl/tls协议传输。它包含证书，卸载，流量转发，负载均衡，页面适配，浏览器适配，refer传递等。保障了传输过程的安全性

15，GET和POST的区别，何时使用POST？

GET：一般用于信息获取，使用URL传递参数，对所发送信息的数量也有限制，一般在2000个字符

POST：一般用于修改服务器上的资源，对所发送的信息没有限制。

GET方式需要使用Request.QueryString来取得变量的值，而POST方式通过Request.Form来获取变量的值，

也就是说Get是通过地址栏来传值，而Post是通过提交表单来传值。

然而，在以下情况中，请使用 POST 请求：

无法使用缓存文件（更新服务器上的文件或数据库）

向服务器发送大量数据（POST 没有数据量限制）

发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

16，ajax的缺点和在IE下的问题？

ajax的缺点

1、ajax不支持浏览器back按钮。

2、安全问题 AJAX暴露了与服务器交互的细节。

3、对搜索引擎的支持比较弱。

4、破坏了程序的异常机制。

5、不容易调试。

IE缓存问题

在IE浏览器下，如果请求的方法是GET，并且请求的URL不变，那么这个请求的结果就会被缓存。解决这个问题的办法可以通过实时改变请求的URL，只要URL改变，就不会被缓存，可以通过在URL末尾添加上随机的时间戳参数('t'= + new Date().getTime())

或者：

open('GET','demo.php?rand=+Math.random()',true);//

Ajax请求的页面历史记录状态问题

可以通过锚点来记录状态，location.hash。让浏览器记录Ajax请求时页面状态的变化。

还可以通过HTML5的history.pushState，来实现浏览器地址栏的无刷新改变

19，说说你对Promise的理解

依照 Promise/A+ 的定义，Promise 有四种状态：

pending: 初始状态, 非 fulfilled 或 rejected.

fulfilled: 成功的操作.

rejected: 失败的操作.

settled: Promise已被fulfilled或rejected，且不是pending

另外， fulfilled 与 rejected 一起合称 settled。

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

Promise 的构造函数

构造一个 Promise，最基本的用法如下：

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

if (...) { // succeed

resolve(result);

} else { // fails

reject(Error(errMessage));

}

});

Promise 实例拥有 then 方法（具有 then 方法的对象，通常被称为 thenable）。它的使用方法如下：

promise.then(onFulfilled, onRejected)

接收两个函数作为参数，一个在 fulfilled 的时候被调用，一个在 rejected 的时候被调用，接收参数就是 future，onFulfilled 对应 resolve, onRejected 对应 reject。

20，前端需要注意哪些SEO

合理的title、description、keywords：搜索对着三项的权重逐个减小，title值强调重点即可，重要关键词出现不要超过2次，而且要靠前，不同页面title要有所不同；description把页面内容高度概括，长度合适，不可 过分堆砌关键词，不同页面description有所不同；keywords列举出重要关键词即可

语义化的HTML代码，符合W3C规范：语义化代码让搜索引擎容易理解网页

重要内容HTML代码放在最前：搜索引擎抓取HTML顺序是从上到下，有的搜索引擎对抓取长度有限制，保证重要内容一定会被抓取

重要内容不要用js输出：爬虫不会执行js获取内容

少用iframe：搜索引擎不会抓取iframe中的内容

非装饰性图片必须加alt

提高网站速度：网站速度是搜索引擎排序的一个重要指标

21，web开发中会话跟踪的方法有哪些

cookie

session

url重写

隐藏input

ip地址

22.什么是Ajax和JSON，它们的优缺点。

Ajax是异步JavaScript和XML，用于在Web页面中实现异步数据交互。

优点：

　 可以使得页面不重载全部内容的情况下加载局部内容，降低数据传输量

　 避免用户不断刷新或者跳转页面，提高用户体验

缺点：

　 对搜索引擎不友好（

　 要实现ajax下的前后退功能成本较大

　 可能造成请求数的增加

　 跨域问题限制

JSON是一种轻量级的数据交换格式，ECMA的一个子集

优点：轻量级、易于人的阅读和编写，便于机器（JavaScript）解析，支持复合数据类型（数组、对象、字符串、数字）

jQuery篇

1，JQuery的源码看过吗？能不能简单概况一下它的实现原理？

jquery源码封装在一个匿名函数的自执行环境中，有助于防止变量的全局污染，然后通过传入window对象参数，可以使window对象作为局部变量使用，好处是当jquery中访问window对象的时候，就不用将作用域链退回到顶层作用域了，从而可以更快的访问window对象。同样，传入undefined参数，可以缩短查找undefined时的作用域链。

(function( window, undefined ) {

//用一个函数域包起来，就是所谓的沙箱

//在这里边var定义的变量，属于这个函数域内的局部变量，避免污染全局

//把当前沙箱需要的外部变量通过函数参数引入进来

//只要保证参数对内提供的接口的一致性，你还可以随意替换传进来的这个参数

window.jQuery = window.$ = jQuery;

})( window );

jquery将一些原型属性和方法封装在了jquery.prototype中，为了缩短名称，又赋值给了jquery.fn，这是很形象的写法。

有一些数组或对象的方法经常能使用到，jQuery将其保存为局部变量以提高访问速度。

jquery实现的链式调用可以节约代码，所返回的都是同一个对象，可以提高代码效率。

jquery的优势就是链式操作，隐式迭代

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？

返回的this指的就是当前操作后的jquery对象，为了实现jquery的链式操作

3，jquery中如何将数组转化为json字符串，然后再转化回来？

使用jquery全局方法$.parseJSON这个方法

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

jQuery中的$.extend浅拷贝，拷贝对象A时，对象B将拷贝A的所有字段，如果字段是内存地址，B将拷贝地址，若果字段是基元类型，B将复制其值。它的缺点是如果你改变了对象B所指向的内存地址，你同时也改变了对象A指向这个地址的字段。

$.extend(a,b)

jQuery中的$.extend深拷贝，这种方式会完全拷贝所有数据，优点是B与A不会相互依赖（A,B完全脱离关联）， 缺点是拷贝的速度更慢，代价更大。

$.extend(true,a,b)

5，jquery.extend 与 jquery.fn.extend的区别？

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

jQuery.extend({

　　 min: function(a, b) { return a < b ? a : b; },

　　 max: function(a, b) { return a > b ? a : b;

});

jQuery.min(2,3); // 2

jQuery.max(4,5); // 5

jQuery.extend(target, object1, [objectN])用一个或多个其他对象来扩展一个对象，返回被扩展的对象。

var settings = { validate: false, limit: 5, name: "foo" };

var options = { validate: true, name: "bar" };

jQuery.extend(settings, options);

结果：settings == { validate: true, limit: 5, name: "bar" }

jQuery.fn.extend(object);

$.fn是什么?

$.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。

查看jQuery代码，就不难发现。

jQuery.fn = jQuery.prototype = {

　　　 init: function( selector, context ) {//....

};

原来 jQuery.fn = jQuery.prototype.

所以，它是对jQuery.prototype进得扩展，就是为jQuery类添加“成员函数”。jQuery类的实例可以使用这个“成员函数”。

jQuery.fn.extend(object); 扩展jQuery对象方法

jQuery.extend 扩展jQuery全局方法

6，jQuery 的队列是如何实现的？队列可以用在哪些地方？

jQuery核心中, 有一组队列控制方法, 这组方法由queue()/dequeue()/clearQueue()三个方法组成, 它对需要连续按序执行的函数的控制可以说是简明自如, 主要应用于animate ()方法, ajax以及其他要按时间顺序执行的事件中.

var \_slideFun = [

function() {

$('.one').delay(500).animate({

top: '+=270px'

},500, \_takeOne);

},

function() {

$('.two').delay(300).animate({

top: '+=270px'

},500, \_takeOne);

}

];

$('#demo').queue('slideList', \_slideFun);

var \_takeOne = function() {

$('#demo').dequeue('slideList');

};

\_takeOne();

7，谈一下Jquery中的bind(),live(),delegate(),on()的区别？

jquery中bind(),live(),delegate()都是基于on实现的，

on是封装了一个兼容的事件绑定方法，在选择元素上绑定一个或多个事件的事件处理函数

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

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

8，JQuery一个对象可以同时绑定多个事件，这是如何实现的？

jquery中事件绑定的函数中传递多个事件参数，执行事件的时候判断执行事件的类型

11，针对 jQuery性能的优化方法？

\*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++) {}

12，Jquery与jQuery UI 有啥区别？

(1) 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。

　 (2) 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) jQuery本身注重于后台，没有漂亮的界面，而jQuery UI则补充了前者的不足，他提供了华丽的展示界面，使人更容易接受。既有强大的后台，又有华丽的前台。jQuery UI是jQuery插件，只不过专指由jQuery官方维护的UI方向的插件。

13，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=function(){} 对象插件

$.function(){} 类插件

14，jQuery和Zepto的区别？各自的使用场景？

zepto主要用在移动设备上，只支持较新的浏览器，好处是代码量比较小，性能也较好。

jquery主要是兼容性好，可以跑在各种pc，移动上，好处是兼容各种浏览器，缺点是代码量大，同时考虑兼容，性能也不够好。

zepto和jQuery选择器实现方法不一样，jQuery使用正则，zepto是使用querySelectAll

zepto针对移动端程序，Zepto还有一些基本的触摸事件可以用来做触摸屏交互，如：

tap事件——tap，singleTap，doubleTap，longTap

Swipe事件——swipe，swipeLeft，swipeRight，swipeUp，swipeDown。

15，Zepto的点透问题如何解决？

1、“点透”是什么

你可能碰到过在列表页面上创建一个弹出层，弹出层有个关闭的按钮，你点了这个按钮关闭弹出层后后，这个按钮正下方的内容也会执行点击事件（或打开链接）。这个被定义为这是一个“点透”现象。

点透的解决方法：

方案一：来得很直接github上有个fastclick可以完美解决https://github.com/ftlabs/fastclick

引入fastclick.js，因为fastclick源码不依赖其他库所以你可以在原生的js前直接加上

1window.addEventListener( "load", function() {

FastClick.attach( document.body );

}, false );

方案二：用touchend代替tap事件并阻止掉touchend的默认行为preventDefault()

方案三：延迟一定的时间(300ms+)来处理事件

16，需求：实现一个页面操作不会整页刷新的网站，并且能在浏览器前进、后退时正确响应。给出你的技术实现方案？

这个可以使用AngelaJs 或者 ReactJs实现单页应用 vue也可以

17，移动端最小触控区域是多大？

移动端最小触控区域 44\*44px ，再小就容易点击不到或者误点

18，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?

每次动画开始的时候先使用stop()函数停止当前未动完的动画

19，移动端的点击事件的有延迟，时间是多久，为什么会有？ 怎么解决这个延时？

click 有 300ms 延迟,为了实现safari的双击事件的设计，浏览器要知道你是不是要双击操作。

20，你从jQuery学到了什么？

首先明白了封装的好处，

链式操作的原理

闭包的好处

21，什么是效果队列？

22，请指出 .get()，[]，eq() 的区别。

.get是jquery中将jquery对象转换为原生对象的方法

[]是采用了获取数组值的方式将jquery对象转换为原生对象的方法

eq()是获取对象列表中的某一个jquery dom对象

23，请指出 .bind()，.live() 和 .delegate() 的区别。

24，请指出 $ 和 $.fn 的区别，或者说出 $.fn 的用途。

$代表的是jquery对象

$.fn是代表的jQuery.prototype

$.fn是用来给jquery对象扩展方法的

25，jQuery取到的元素和原生Js取到的元素有什么区别

jQuery取到的元素是包含原生dom对象，和jQuery扩展的方法

26，jQuery的联级有什么好处

27，原生JS的window.onload与Jquery的$(document).ready(function(){})有什么不同？如何用原生JS实现Jq的ready方法？

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

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

function ready(fn){

if(document.addEventListener) { //标准浏览器

document.addEventListener('DOMContentLoaded', function() {

//注销事件, 避免反复触发

document.removeEventListener('DOMContentLoaded',arguments.callee, false);

fn(); //执行函数

}, false);

}else if(document.attachEvent) { //IE

document.attachEvent('onreadystatechange', function() {

if(document.readyState == 'complete') {

document.detachEvent('onreadystatechange', arguments.callee);

fn(); //函数执行

}

});

}

};

28，jquery中常用的方法说出几个来

移动开发篇

你怎么看待Web App 、hybrid App、Native App？

你移动端前端开发的理解？（和 Web 前端开发的主要区别是什么？）

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、

简单描述一下你做过的移动APP项目研发流程？

你认为怎样才是全端工程师（Full Stack developer）？

什么是Retina 显示屏，带来了什么问题？

retina：一种具备超高像素密度的液晶屏，同样大小的屏幕上显示的像素点由1个变为多个，如在同样带下的屏幕上，苹果设备的retina显示屏中，像素点1个变为4个

在高清显示屏中的位图被放大，图片会变得模糊，因此移动端的视觉稿通常会设计为传统PC的2倍

那么，前端的应对方案是：

设计稿切出来的图片长宽保证为偶数，并使用backgroud-size把图片缩小为原来的1/2

//例如图片宽高为：200px\*200px，那么写法如下

.css{width:100px;height:100px;background-size:100px 100px;}

其它元素的取值为原来的1/2，例如视觉稿40px的字体，使用样式的写法为20px

.css{font-size:20px}

ios系统中元素被触摸时产生的半透明灰色遮罩怎么去掉

ios用户点击一个链接，会出现一个半透明灰色遮罩, 如果想要禁用，可设置-webkit-tap-highlight-color的alpha值为0，也就是属性值的最后一位设置为0就可以去除半透明灰色遮罩

a,button,input,textarea{-webkit-tap-highlight-color: rgba(0,0,0,0;)}

部分android系统中元素被点击时产生的边框怎么去掉

android用户点击一个链接，会出现一个边框或者半透明灰色遮罩, 不同生产商定义出来额效果不一样，可设置-webkit-tap-highlight-color的alpha值为0去除部分机器自带的效果

a,button,input,textarea{

-webkit-tap-highlight-color: rgba(0,0,0,0;)

-webkit-user-modify:read-write-plaintext-only;

}

-webkit-user-modify有个副作用，就是输入法不再能够输入多个字符

另外，有些机型去除不了，如小米2

对于按钮类还有个办法，不使用a或者input标签，直接用div标签

webkit表单元素的默认外观怎么重置

通用

.css{-webkit-appearance:none;}

伪元素改变number类型input框的默认样式

input[type=number]::-webkit-textfield-decoration-container {

background-color: transparent;

}

input[type=number]::-webkit-inner-spin-button {

-webkit-appearance: none;

}

input[type=number]::-webkit-outer-spin-button {

-webkit-appearance: none;

}

webkit表单输入框placeholder的颜色值能改变么

input::-webkit-input-placeholder{color:#AAAAAA;}

input:focus::-webkit-input-placeholder{color:#EEEEEE;}

webkit表单输入框placeholder的文字能换行么

ios可以，android不行~

在textarea标签下都可以换行~

禁止ios 长按时不触发系统的菜单，禁止ios&android长按时下载图片

.css{-webkit-touch-callout: none}

禁止ios和android用户选中文字

.css{-webkit-user-select:none}

打电话发短信写邮件怎么实现

打电话： <a href="tel:0755-10086">打电话给:0755-10086</a>

发短信，winphone系统无效 <a href="sms:10086">发短信给: 10086</a>

写邮件： <a href=“mailto:peun@foxmail.com">peun@foxmail.com</a>

模拟按钮hover效果

移动端触摸按钮的效果，可明示用户有些事情正要发生，是一个比较好体验，但是移动设备中并没有鼠标指针，使用css的hover并不能满足我们的需求，还好国外有个激活移动端css的active效果。

1，直接在body上添加ontouchstart，同样可激活移动端css的active效果，比较推荐这种方式（兼容性ios5+、部分android 4+、winphone 8）

2，要做到全兼容的办法，可通过绑定ontouchstart和ontouchend来控制按钮的类名

audio元素和video元素在ios和andriod中无法自动播放

应对方案：触屏即播

$('html').one('touchstart',function(){

audio.play()

})

消除transition闪屏

网络都是这么写的，但我并没有测试出来

.css{

/\*设置内嵌的元素在 3D 空间如何呈现：保留 3D\*/

-webkit-transform-style: preserve-3d;

/\*（设置进行转换的元素的背面在面对用户时是否可见：隐藏）\*/

-webkit-backface-visibility: hidden;

}

开启硬件加速

1，解决页面闪白

2，保证动画流畅

.css {

-webkit-transform: translate3d(0, 0, 0);

-moz-transform: translate3d(0, 0, 0);

-ms-transform: translate3d(0, 0, 0);

transform: translate3d(0, 0, 0);

}

取消input在ios下，输入的时候英文首字母的默认大写

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

android 上去掉语音输入按钮

input::-webkit-input-speech-button {display: none}

android 2.3 bug

@-webkit-keyframes 需要以0%开始100%结束，0%的百分号不能去掉

after和before伪类无法使用动画animation

border-radius不支持%单位

translate百分比的写法和scale在一起会导致失效，例如-webkit-transform: translate(-50%,-50%) scale(-0.5, 1)

android 4.x bug

三星 Galaxy S4中自带浏览器不支持border-radius缩写

同时设置border-radius和背景色的时候，背景色会溢出到圆角以外部分

部分手机(如三星)，a链接支持鼠标:visited事件，也就是说链接访问后文字变为紫色

android无法同时播放多音频audio

设计高性能CSS3动画的几个要素

尽可能地使用合成属性transform和opacity来设计CSS3动画，不使用position的left和top来定位

利用translate3D开启GPU加速

fixed bug

ios下fixed元素容易定位出错，软键盘弹出时，影响fixed元素定位

android下fixed表现要比iOS更好，软键盘弹出时，不会影响fixed元素定位

ios4下不支持position:fixed

解决方案

可用isroll.js，暂无完美方案

播放视频不全屏

<!--

1.目前只有ios7+、winphone8+支持自动播放

2.支持Airplay的设备（如：音箱、Apple TV)播放

x-webkit-airplay="true"

3.播放视频不全屏，ios7+、winphone8+支持，部分android4+支持（含华为、小米、魅族）

webkit-playsinline="true"

-->

<video x-webkit-airplay="true" webkit-playsinline="true" preload="auto" autoplay src="http://"></video>

体验demo：http://1.peunzhang.sinaapp.com/demo/video/index.html

H5页面窗口自动调整到设备宽度，并禁止用户缩放页面

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

忽略将页面中的数字识别为电话号码

<meta name="format-detection" content="telephone=no" />

忽略Android平台中对邮箱地址的识别

<meta name="format-detection" content="email=no" />

当网站添加到主屏幕快速启动方式，可隐藏地址栏，仅针对ios的safari

<meta name="apple-mobile-web-app-capable" content="yes" />

<!-- ios7.0版本以后，safari上已看不到效果 —>

判断是否为iPhone：

// 判断是否为 iPhone ：

function isAppleMobile() {

return (navigator.platform.indexOf('iPad') != -1);

}

移动H5前端性能优化指南

1. PC优化手段在Mobile侧同样适用

2. 在Mobile侧我们提出三秒种渲染完成首屏指标

3. 基于第二点，首屏加载3秒完成或使用Loading

4. 基于联通3G网络平均338KB/s(2.71Mb/s)，所以首屏资源不应超过1014KB

5. Mobile侧因手机配置原因，除加载外渲染速度也是优化重点

6. 基于第五点，要合理处理代码减少渲染损耗

7. 基于第二、第五点，所有影响首屏加载和渲染的代码应在处理逻辑中后置

8. 加载完成后用户交互使用时也需注意性能

优化指南

[加载优化]

加载过程是最为耗时的过程，可能会占到总耗时的80%时间，因此是优化的重点

· 减少HTTP请求

因为手机浏览器同时响应请求为4个请求（Android支持4个，iOS 5后可支持6个），所以要尽量减少页面的请求数，首次加载同时请求数不能超过4个

a) 合并CSS、JavaScript

b) 合并小图片，使用雪碧图

· 缓存

使用缓存可以减少向服务器的请求数，节省加载时间，所以所有静态资源都要在服务器端设置缓存，并且尽量使用长Cache（长Cache资源的更新可使用时间戳）

a) 缓存一切可缓存的资源

b) 使用长Cache（使用时间戳更新Cache）

c) 使用外联式引用CSS、JavaScript

· 压缩HTML、CSS、JavaScript

减少资源大小可以加快网页显示速度，所以要对HTML、CSS、JavaScript等进行代码压缩，并在服务器端设置GZip

a) 压缩（例如，多余的空格、换行符和缩进）

b) 启用GZip

· 无阻塞

写在HTML头部的JavaScript（无异步），和写在HTML标签中的Style会阻塞页面的渲染，因此CSS放在页面头部并使用Link方式引入，避免在HTML标签中写Style，JavaScript放在页面尾

部或使用异步方式加载

· 使用首屏加载

首屏的快速显示，可以大大提升用户对页面速度的感知，因此应尽量针对首屏的快速显示做优化

· 按需加载

将不影响首屏的资源和当前屏幕资源不用的资源放到用户需要时才加载，可以大大提升重要资源的显示速度和降低总体流量

PS：按需加载会导致大量重绘，影响渲染性能

a) LazyLoad

b) 滚屏加载

c) 通过Media Query加载

· 预加载

大型重资源页面（如游戏）可使用增加Loading的方法，资源加载完成后再显示页面。但Loading时间过长，会造成用户流失

对用户行为分析，可以在当前页加载下一页资源，提升速度

a) 可感知Loading(如进入空间游戏的Loading)

b) 不可感知的Loading（如提前加载下一页）

· 压缩图片

图片是最占流量的资源，因此尽量避免使用他，使用时选择最合适的格式（实现需求的前提下，以大小判断），合适的大小，然后使用智图压缩，同时在代码中用Srcset来按需显示

PS：过度压缩图片大小影响图片显示效果

a) 使用智图（ http://zhitu.isux.us/ ）

b) 使用其它方式代替图片(1. 使用CSS3 2. 使用SVG 3. 使用IconFont)

c) 使用Srcset

d) 选择合适的图片(1. webP优于JPG 2. PNG8优于GIF)

e) 选择合适的大小（1. 首次加载不大于1014KB 2. 不宽于640（基于手机屏幕一般宽度））

· 减少Cookie

Cookie会影响加载速度，所以静态资源域名不使用Cookie

· 避免重定向

重定向会影响加载速度，所以在服务器正确设置避免重定向

· 异步加载第三方资源

第三方资源不可控会影响页面的加载和显示，因此要异步加载第三方资源

[脚本执行优化]

脚本处理不当会阻塞页面加载、渲染，因此在使用时需当注意

· CSS写在头部，JavaScript写在尾部或异步

· 避免图片和iFrame等的空Src

空Src会重新加载当前页面，影响速度和效率

· 尽量避免重设图片大小

重设图片大小是指在页面、CSS、JavaScript等中多次重置图片大小，多次重设图片大小会引发图片的多次重绘，影响性能

· 图片尽量避免使用DataURL

DataURL图片没有使用图片的压缩算法文件会变大，并且要解码后再渲染，加载慢耗时长

[CSS优化]

· 尽量避免写在HTML标签中写Style属性

· 避免CSS表达式

CSS表达式的执行需跳出CSS树的渲染，因此请避免CSS表达式

· 移除空的CSS规则

空的CSS规则增加了CSS文件的大小，且影响CSS树的执行，所以需移除空的CSS规则

· 正确使用Display的属性

Display属性会影响页面的渲染，因此请合理使用

a) display:inline后不应该再使用width、height、margin、padding以及float

b) display:inline-block后不应该再使用float

c) display:block后不应该再使用vertical-align

d) display:table-\*后不应该再使用margin或者float

· 不滥用Float

Float在渲染时计算量比较大，尽量减少使用  
 · 不滥用Web字体

Web字体需要下载，解析，重绘当前页面，尽量减少使用

· 不声明过多的Font-size

过多的Font-size引发CSS树的效率

· 值为0时不需要任何单位

为了浏览器的兼容性和性能，值为0时不要带单位

· 标准化各种浏览器前缀

a) 无前缀应放在最后

b) CSS动画只用 （-webkit- 无前缀）两种即可

c) 其它前缀为 -webkit- -moz- -ms- 无前缀 四种，（-o-Opera浏览器改用blink内核，所以淘汰）

· 避免让选择符看起来像正则表达式 高级选择器执行耗时长且不易读懂，避免使用

[JavaScript执行优化]

· 减少重绘和回流

a) 避免不必要的Dom操作

b) 尽量改变Class而不是Style，使用classList代替className

c) 避免使用document.write

d) 减少drawImage

· 缓存Dom选择与计算

每次Dom选择都要计算，缓存他

· 缓存列表.length

每次.length都要计算，用一个变量保存这个值

· 尽量使用事件代理，避免批量绑定事件

· 尽量使用ID选择器

ID选择器是最快的

· TOUCH事件优化

使用touchstart、touchend代替click，因快影响速度快。但应注意Touch响应过快，易引发误操作

[渲染优化]

· HTML使用Viewport

Viewport可以加速页面的渲染，请使用以下代码

<meta name=”viewport” content=”width=device-width, initial-scale=1″>

· 减少Dom节点

Dom节点太多影响页面的渲染，应尽量减少Dom节点

· 动画优化

a) 尽量使用CSS3动画

b) 合理使用requestAnimationFrame动画代替setTimeout

c) 适当使用Canvas动画 5个元素以内使用css动画，5个以上使用Canvas动画（iOS8可使用webGL）

· 高频事件优化

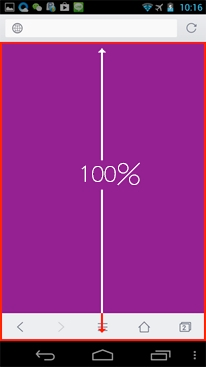
Touchmove、Scroll 事件可导致多次渲染

a) 使用requestAnimationFrame监听帧变化，使得在正确的时间进行渲染

b) 增加响应变化的时间间隔，减少重绘次数

· GPU加速

CSS中以下属性（CSS3 transitions、CSS3 3D transforms、Opacity、Canvas、WebGL、Video）来触发GPU渲染，请合理使用（PS：过渡使用会引发手机过耗电增加）

页面100%高度包含地址栏高度，当地址栏存在时，会有部分内容被隐藏，如下图所示：

重置<html>高度；document.documentElement.style.height = window.innerHeight + ‘px’;

如何阻止浏览器默认滚动事件更好？

默认事件禁止touchmove比touchstart更好，否则还需要单独处理<a><input>等问题。

document.documentElement.addEventListener('touchmove', function(e){

e.preventDefault();

});

CSS动画与Canvas动画性能优劣分析

问题：

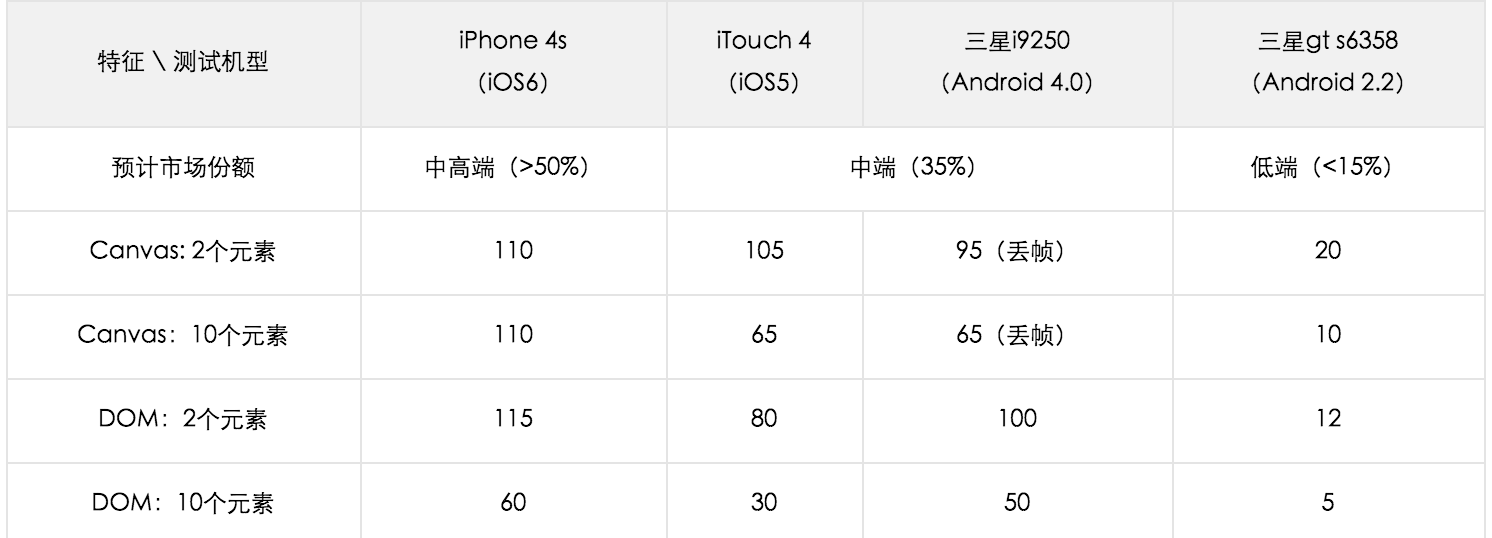
目前移动端动画，特别在H5游戏与强交互性网站应用非常普遍，常用的实现方式是CSS（硬件加速）与Canvas，但缺乏完整的性能分析数据，我们在选择动画实现方式时还是存在纠结。

解决：

选用高中低不同配置不同系统的4款设备，进行了一轮测试：



DOM：2个元素 DOM：10个元素



测试数据：

一般60以上较为顺畅。

3）、测试结论：

a、CSS动画更为流畅、但内存占用过高，动画元素在5个以内更为推荐；

b、Canvas动画存在丢帧现象，这一现象在android中低端手机中表现更为明显；

c、5个以内动画元素，选用CSS动画，80%的设备帧频可达80以上。

在做移动端的时候你时如何调试的

其他

除了前端以外还了解什么其它技术么？你最最厉害的技能是什么？

你用的得心应手用的熟练地编辑器&开发环境是什么样子？

对前端工程师这个职位是怎么样理解的？它的前景会怎么样？

前端是最贴近用户的程序员，比后端、数据库、产品经理、运营、安全都近。

1、实现界面交互

2、提升用户体验

3、有了Node.js，前端可以实现服务端的一些事情

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，

参与项目，快速高质量完成实现效果图，精确到1px；

与团队成员，UI设计，产品经理的沟通；

做好的页面结构，页面重构和用户体验；

处理hack，兼容、写出优美的代码格式；

针对服务器的优化、拥抱最新前端技术。

从我短暂的前端开发之路的过程中来看，前端开发的发展越来越贴近「传统」的后端程序员，也许在未来的某个时间点上，前端和后端的隔阂将彻底的被打破，浏览器上的「工匠」将统称为web开发者。

你对加班的看法？

平时如何管理你的项目？

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

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）；

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）；

页面进行标注（例如 页面 模块 开始和结束）；

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

JS 分文件夹存放 命名以该JS功能为准的英文翻译。

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

当团队人手不足，把功能代码写完已经需要加班的情况下，你会做前端代码的测试吗

会，一般都是先开法功能做好了之后会粗略的测试一下，有一些小问题的话能改就马上改了，大问题的话，我可能心情立马降到0点没什么心情改，就先放下第二天在改

说说最近最流行的一些东西吧？常去哪些网站？

ES6\WebAssembly\Node\MVVM\Web Components\React\React Native\Webpack 组件化

知道什么是SEO并且怎么优化么? 知道各种meta data的含义么?

合理的title、description、keywords：搜索对着三项的权重逐个减小，title值强调重点即可，重要关键词出现不要超过2次，而且要靠前，不同页面title要有所不同；description把页面内容高度概括，长度合适，不可过分堆砌关键词，不同页面description有所不同；keywords列举出重要关键词即可

语义化的HTML代码，符合W3C规范：语义化代码让搜索引擎容易理解网页

重要内容HTML代码放在最前：搜索引擎抓取HTML顺序是从上到下，有的搜索引擎对抓取长度有限制，保证重要内容一定会被抓取

重要内容不要用js输出：爬虫不会执行js获取内容

少用iframe：搜索引擎不会抓取iframe中的内容

非装饰性图片必须加alt

提高网站速度：网站速度是搜索引擎排序的一个重要指标

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？

在之前的企业我觉得我是处于一个路由器的作用，在公司里面得和产品沟通（功能的合理性），设计沟通（页面效果实现），后台沟通（数据逻辑如何实现更加简单），测试还的沟通（很多测试会把浏览器的bug都给测试出来。。。）

你认为怎样才是全端工程师（Full Stack developer）？

我对全端工程师的定义是：掌握多种技能，并能利用多种技能独立完成产品的人

会后端，会前端，会安卓，会ios，还懂得设计交互，能一个人干3-5个人的活，而且比这3-5个干的好

介绍一个你最得意的作品吧？

你有自己的技术博客吗，用了哪些技术？

对前端安全有什么看法？

项目中遇到国哪些印象深刻的技术难题，具体是什么问题，怎么解决？。

这个问题很常见，有没有遇到过很不常见的问题？比如在网上根本搜不到解决方法的？

最近在学什么东西？

你的优点是什么？缺点是什么？

如何管理前端团队?

在管理前端团队方面的话我还没有一定的经验，我说一下我的见解吧

首先明确团队中每个人负责的模块（根据个人技术水平来分话）

明确产品前端技术负责人（他负责pc,他负责移动）

最重要的就是提升前端团队的技术水平（定期分享前端技术，大家共同进步）

最近在学什么？能谈谈你未来3，5年给自己的规划吗？

前端和后端程序员应该如何合作？

在之前公司里，前后端在开发和调试过程中的交流比跟家里的媳妇儿交流的还要多

我已经经历过的开发过程中，前后端的配合和讨论多集中于：

数据交换时的约定

什么样的逻辑可以拿到前端进行处理，在安全可靠健壮的同时减少服务器端的压力。

数据交换方式（json,字符串,跳转页面时的参数传递等）

在JavaScript面向对象方面，你有什么体会和实践？

任何技术都不应该为了实现而实现。在刚开始接触面向对象的时候，想把所有代码都写成面向对象的。现在想想，有一些功能和逻辑如果通过面向对象来实现，效率不一定会提高，甚至会降低很多。

继承：javascript的继承使用链式继承，一个类与其父类和子类的关系都是通过原型链来追溯的。javascript中的函数是引用类型，这就说明如果一个父类的多个子类中的某一个子类通过引用修改了父类的函数，所有的子类都会受到影响。

多态：Javascript是不支持多态的，但是我们可以通过一些设计模式来模仿多态，比如工厂方法，在函数入口判断传入参数调用不同的方法。调用某一个类的方法，会在原型链中该类的位置开始，逐渐向父类的方向查找，所以「重载」也就可以实现了，如果子类有该方法，就会调用子类的方法，若没有，会向上查找。

封装：函数的作用域很好的保证了函数内部的变量不会被外部访问到。私有状态的变量和属性可以通过将变量闭包在构造函数内完成。这样，私有状态的变量会在每个对象上被创建，而不是从原型链进行继承。

AMD（Modules/Asynchronous-Definition）、CMD（Common Module Definition）规范区别？

Asynchronous Module Definition，异步模块定义，所有的模块将被异步加载，模块加载不影响后面语句运行。所有依赖某些模块的语句均放置在回调函数中。

区别：

1. 对于依赖的模块，AMD 是提前执行，CMD 是延迟执行。不过 RequireJS 从 2.0 开始，也改成可以延迟执行（根据写法不同，处理方式不同）。CMD 推崇 as lazy as possible.

2. CMD 推崇依赖就近，AMD 推崇依赖前置。看代码：

// CMD

define(function(require, exports, module) {

var a = require('./a')

a.doSomething()

// 此处略去 100 行

var b = require('./b') // 依赖可以就近书写

b.doSomething()

// ...

})

// AMD 默认推荐

define(['./a', './b'], function(a, b) { // 依赖必须一开始就写好

a.doSomething()

// 此处略去 100 行

b.doSomething()

// ...

})

requireJS的核心原理是什么？（如何动态加载的？如何避免多次加载的？如何 缓存的？）

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?

原来公司工作流程是怎么样的，如何与其他人协作的？如何夸部门合作的？

简述一下 Handlebars 的基本用法？

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？

想问公司的问题？

你是如何接触前端的？

公司的薪资结构是什么样子的？

是否有设计过通用的组件

请设计一个 Dialog（弹出层） / Suggestion（自动完成） / Slider（图片轮播） 等组件

你会提供什么接口？

调用过程是怎样的？可能会遇到什么细节问题？

你用bootstrap中最多的组件是什么？

删格化布局， 轮播组件，表格组件，导航组件，表单组件，按钮组件。。。

必问人资题

请你自我介绍一下

思路：

　　 1、这是面试的必考题目。

　　 2、介绍内容要与个人简历相一致。

　　 3、表述方式上尽量口语化。

　　 4、要切中要害，不谈无关、无用的内容。

　　 5、条理要清晰，层次要分明。

　　 6、事先最好以文字的形式写好背熟。

7，技术要点

8，项目经验

你有什么业余爱好？

思路：

　　 1、 业余爱好能在一定程度上反映应聘者的性格、观念、心态，这是招聘单位问该问题的主要原因。

　　 2、 最好不要说自己没有业余爱好。

　　 3、 不要说自己有那些庸俗的、令人感觉不好的爱好。

　　 4、 最好不要说自己仅限于读书、听音乐、上网，否则可能令面试官怀疑应聘者性格孤僻。

　　 5、 最好能有一些户外的业余爱好来“点缀”你的形象。

谈谈你的缺点

　　思路：

　　 1、 不宜说自己没缺点。

　　 2、 不宜把那些明显的优点说成缺点。

　　 3、 不宜说出严重影响所应聘工作的缺点。

　　 4、 不宜说出令人不放心、不舒服的缺点。

　　 5、 可以说出一些对于所应聘工作“无关紧要”的缺点，甚至是一些表面上看是缺点，从工作的角度看却是优点的缺点。

您在前一家公司的离职原因是什么？

　思路：

　　 1、 最重要的是：应聘者要使找招聘单位相信，应聘者在过往的单位的“离职原因”在此家招聘单位里不存在。

2、 避免把“离职原因”说得太详细、太具体。

　　 3、 不能掺杂主观的负面感受，如“太幸苦”、“人际关系复杂”、“管理太混乱”、“公司不重视人才”、“公司排斥我们某某的员工”等。

　　 4、 但也不能躲闪、回避，如“想换换环境”、“个人原因”等。

　　 5、 不能涉及自己负面的人格特征，如不诚实、懒惰、缺乏责任感、不随和等。

　　 6、 尽量使解释的理由为应聘者个人形象添彩。

　　 7、 如“我离职是因为这家公司倒闭。我在公司工作了三年多，有较深的感情。从去年始，由于市场形势突变，公司的局面急转直下。到眼下这一步我觉得很遗憾，但还要面对显示，重新寻找能发挥我能力的舞台。” 同一个面试问题并非只有一个答案，而同一个答案并不是在任何面试场合都有效，关键在于应聘者掌握了规律后，对面试的具体情况进行把握，有意识地揣摩面试官提出问题的心理背景，然后投其所好。

css命名规范

命名规则遵循一个统一的规则能够方便团队协作和后期维护

来自快切网工作中使用的命名规范。

id和class命名采用该版块的英文单词或组合命名，并第一个单词小写，第二个单词首个字母大写，如:QuickCss（最新产品/Quick+Css)

CSS样式表各区块用注释说明

尽量使用英文命名原则

尽量不加中杠和下划线

尽量不缩写，除非一看就明白的单词

css命名

好的css命名是检验一个css工作者是否合格的一项基本指标。

头：header

内容：content/container

尾：footer

导航：nav

侧栏：sidebar

栏目：col

页面外围控制整体布局宽度：wrapper

左右中：left right center

登录条：loginBar

标志：logo

广告：banner

页面主体：main

热点：hot

新闻：news

下载：download

子导航：subNav

菜单：menu

子菜单：subMenu

搜索：search

友情链接：friendLink

页脚：footer

版权：copyright

滚动：scroll

内容：content

标签页：tab

文章列表：list

提示信息：msg

小技巧：tips

栏目标题：title

加入：joinus

指南：guide

服务：service

注册：register

状态：status

投票：vote

合作伙伴：partner

(二)注释的写法:

/\* Footer \*/

内容区

/\* End Footer \*/

(三)id的命名:

(1)页面结构

容器: container

页头：header

内容：content/container

页面主体：main

页尾：footer

导航：nav

侧栏：sidebar

栏目：col

页面外围控制整体布局宽度：wrapper

左右中：left right center

(2)导航

导航：nav

主导航：mainNav

子导航：subNav

顶导航：topNav

边导航：sidebar

左导航：leftSidebar

右导航：rightSidebar

菜单：menu

子菜单：subMenu

标题: title

摘要: summary

(3)功能

标志：logo

广告：banner

登陆：login

登录条：loginBar

注册：register

搜索：search

功能区：shop

标题：title

加入：joinus

状态：status

按钮：btn

滚动：scroll

标签页：tab

文章列表：list

提示信息：msg

当前的: current

小技巧：tips

图标: icon

注释：note

指南：guide

服务：service

热点：hot

新闻：news

下载：download

投票：vote

合作伙伴：partner

友情链接：link

版权：copyright

(四)class的命名:

(1)标题栏样式,使用"类别+功能"的方式命名,如

.barNews { }

.barProduct { }

(2)模块结构css定议

模块标题 .module

模块标题 .moduleHead

模块包装 .moduleWrap

模块内容 .moduleContent

注意事项:

1.一律小写;

2.尽量用英文;

3.不加中杠和下划线;

4.尽量不缩写，除非一看就明白的单词.

主要的 master.css

模块 module.css

基本共用 base.css

布局，版面 layout.css

面试问题收集

跨域有几种方法？

跨域的话我研究过的有两种方法，jsonp和CORS两种，

jsonp的话以前比较常用一些，主要实现原理就是通过引用js的方法来获取跨域资源，通过执行js函数的方法调用回调成功的函数，

cors的话这个比较简单一些，但是有一些兼容问题，只支持ie10+ 和现代浏览器适合用于移动端，实现原理就是在后端加上header头信息Access-Control-Allow-Origin: ＊，也可以指定域名访问Access-Control-Allow-Origin: <http://api.qiutc.me>

其他的方法暂时还没有研究过

jsonp的原理

jsonp的实现原理就是，jquery中动态创建一个script元素，然后给script的scr属性赋值为ajax请求接口地址，在接口地址的后面添加一个callback的参数，参数的值就是jquery中动态生成的函数的函数名称，后台更具参数返回值

返回的结构是 方法名（json值） 成功后就相当于执行了一个js函数，就是ajax中success的成功后的回调函数

请解释一下 JavaScript 的同源策略。

同源策略是客户端脚本（尤其是Javascript）的重要的安全度量标准。它最早出自Netscape Navigator2.0，其目的是防止某个文档或脚本从多个不同源装载。

它的精髓很简单：它认为自任何站点装载的信赖内容是不安全的。当被浏览器半信半疑的脚本运行在沙箱时，它们应该只被允许访问来自同一站点的资源，而不是那些来自其它站点可能怀有恶意的资源。

这里的同源指的是：同协议，同域名和同端口

前端性能优化的方法？

（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，显示比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。

CSS优化、提高性能的方法有哪些？

将样式表放到页面顶部

不使用CSS表达式

不使用@import

不使用IE的Filter

针对 jQuery性能的优化方法？

\*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++) {}

移动H5前端性能优化指南

1. PC优化手段在Mobile侧同样适用

2. 在Mobile侧我们提出三秒种渲染完成首屏指标

3. 基于第二点，首屏加载3秒完成或使用Loading

4. 基于联通3G网络平均338KB/s(2.71Mb/s)，所以首屏资源不应超过1014KB

5. Mobile侧因手机配置原因，除加载外渲染速度也是优化重点

6. 基于第五点，要合理处理代码减少渲染损耗

7. 基于第二、第五点，所有影响首屏加载和渲染的代码应在处理逻辑中后置

8. 加载完成后用户交互使用时也需注意性能

JQuery的源码看过吗？能不能简单概况一下它的实现原理？

jquery源码封装在一个匿名函数的自执行环境中，有助于防止变量的全局污染，然后通过传入window对象参数，可以使window对象作为局部变量使用，好处是当jquery中访问window对象的时候，就不用将作用域链退回到顶层作用域了，从而可以更快的访问window对象。同样，传入undefined参数，可以缩短查找undefined时的作用域链。

(function( window, undefined ) {

//用一个函数域包起来，就是所谓的沙箱

//在这里边var定义的变量，属于这个函数域内的局部变量，避免污染全局

//把当前沙箱需要的外部变量通过函数参数引入进来

//只要保证参数对内提供的接口的一致性，你还可以随意替换传进来的这个参数

window.jQuery = window.$ = jQuery;

})( window );

jquery在jquery的方法中实例化了一个init方法，这个init方法是jquery原型里面的一个方法，每次$()调用的时候就会实例化一个init方法，而init方法的原型＝jquery的原型，所以我们没错调用其实还是相当于实例化jquery，只不过他进行了一个封装，减少了调用时候的步骤

jquery将一些原型属性和方法封装在了jquery.prototype中，为了缩短名称，又赋值给了jquery.fn，这是很形象的写法。

有一些数组或对象的方法经常能使用到，jQuery将其保存为局部变量以提高访问速度。

jquery实现的链式调用可以节约代码，所返回的都是同一个对象，可以提高代码效率。

jquery的优势就是链式操作，隐式迭代

如何区别object和arrey

object.protopyte.toString();

locationStorage 和 cookie

jQuery.extend 函数详解

为jQuery类添加类方法，可以理解为添加静态方法

jQuery.extend({

min: function(a, b) { return a < b ? a : b; },

max: function(a, b) { return a > b ? a : b; }

});

jQuery.min(2,3); // 2

jQuery.max(4,5); // 5

用一个或多个其他对象来扩展一个对象，返回被扩展的对象

var settings = { validate: false, limit: 5, name: "foo" };

var options = { validate: true, name: "bar" };

jQuery.extend(settings, options);

结果：settings == { validate: true, limit: 5, name: "bar" }

AngularJs面试题

ng-if 跟 ng-show/hide 的区别有哪些？

第一点区别是，ng-if 在后面表达式为 true 的时候才创建这个 dom 节点，ng-show 是初始时就创建了，用 display:block 和 display:none 来控制显示和不显示。

第二点区别是，ng-if 会（隐式地）产生新作用域，ng-switch 、 ng-include 等会动态创建一块界面的也是如此。

这样会导致，在 ng-if 中用基本变量绑定 ng-model，并在外层 div 中把此 model 绑定给另一个显示区域，内层改变时，外层不会同步改变，因为此时已经是两个变量了。

<p>{{name}}</p>

<div ng-if="true">

<input type="text" ng-model="name">

</div>

ng-show 不存在此问题，因为它不自带一级作用域。

避免这类问题出现的办法是，始终将页面中的元素绑定到对象的属性（data.x）而不是直接绑定到基本变量（x）上。

ng-repeat迭代数组的时候，如果数组中有相同值，会有什么问题，如何解决？

会提示 Duplicates in a repeater are not allowed. 加 track by $index 可解决。当然，也可以 trace by 任何一个普通的值，只要能唯一性标识数组中的每一项即可（建立 dom 和数据之间的关联）。

ng-click 中写的表达式，能使用 JS 原生对象上的方法吗？

不止是 ng-click 中的表达式，只要是在页面中，都不能直接调用原生的 JS 方法，因为这些并不存在于与页面对应的 Controller 的 $scope 中。

{{now | 'yyyy-MM-dd'}} 这种表达式里面，竖线和后面的参数通过什么方式可以自定义？

filter，格式化数据，接收一个输入，按某规则处理，返回处理结果。

ng 内置的 filter 有九种：

date（日期）

currency（货币）

limitTo（限制数组或字符串长度）

orderBy（排序）

lowercase（小写）

uppercase（大写）

number（格式化数字，加上千位分隔符，并接收参数限定小数点位数）

filter（处理一个数组，过滤出含有某个子串的元素）

json（格式化 json 对象）

filter 有两种使用方法，一种是直接在页面里：

<p>{{now | date : 'yyyy-MM-dd'}}</p>

另一种是在 js 里面用：

// $filter('过滤器名称')(需要过滤的对象, 参数1, 参数2,...)

$filter('date')(now, 'yyyy-MM-dd hh:mm:ss’);

自定义 filter

// 形式

app.filter('过滤器名称',function(){

return function(需要过滤的对象,过滤器参数1,过滤器参数2,...){

//...做一些事情

return 处理后的对象;

}

});

// 栗子

app.filter('timesFilter', function(){

return function(item, times){

var result = '';

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

result += item;

}

return result;

}

})

factory、service 和 provider 是什么关系？

factory：把 service 的方法和数据放在一个对象里，并返回这个对象

service：通过构造函数方式创建 service，返回一个实例化对象

provider：创建一个可通过 config 配置的 service，$get 中返回的，就是用 factory 创建 service 的内容

从底层实现上来看，service 调用了 factory，返回其实例；factory 调用了 provider，返回其 $get 中定义的内容。factory 和 service 功能类似，只不过 factory 是普通 function，可以返回任何东西（return 的都可以被访问，所以那些私有变量怎么写，你懂的）；service 是构造器，可以不返回（绑定到 this 的都可以被访问）；provider 是加强版 factory，返回一个可配置的 factory。

angular 的数据绑定采用什么机制？详述原理

脏检查机制。

双向数据绑定是 AngularJS 的核心机制之一。当 view 中有任何数据变化时，会更新到 model ，当 model 中数据有变化时，view 也会同步更新，显然，这需要一个监控。

原理就是，Angular 在 scope 模型上设置了一个 监听队列，用来监听数据变化并更新 view 。每次绑定一个东西到 view 上时 AngularJS 就会往 $watch 队列里插入一条 $watch，用来检测它监视的 model 里是否有变化的东西。当浏览器接收到可以被 angular context 处理的事件时，$digest 循环就会触发，遍历所有的 $watch，最后更新 dom。

angular 中控制器之间如何通信？

一个 angular 应用应当如何良好地分层？

目录结构的划分

对于小型项目，可以按照文件类型组织，比如：

css

js

controllers

models

services

filters

templates

但是对于规模较大的项目，最好按业务模块划分，比如：

css

modules

account

controllers

models

services

filters

templates

disk

controllers

models

services

filters

templates

modules 下最好再有一个 common 目录来存放公共的东西。

逻辑代码的拆分

作为一个 MVVM 框架，Angular 应用本身就应该按照 模型，视图模型（控制器），视图来划分。

这里逻辑代码的拆分，主要是指尽量让 controller 这一层很薄。提取共用的逻辑到 service 中 （比如后台数据的请求，数据的共享和缓存，基于事件的模块间通信等），提取共用的界面操作到 directive 中（比如将日期选择、分页等封装成组件等），提取共用的格式化操作到 filter 中等等。

在复杂的应用中，也可以为实体建立对应的构造函数，比如硬盘（Disk）模块，可能有列表、新建、详情这样几个视图，并分别对应的有 controller，那么可以建一个 Disk 构造函数，里面完成数据的增删改查和验证操作，有跟 Disk 相关的 controller，就注入 Disk 构造器并生成一个实例，这个实例就具备了增删改查和验证方法。这样既层次分明，又实现了复用（让 controller 层更薄了）。

angular 应用常用哪些路由库，各自的区别是什么？

Angular1.x 中常用 ngRoute 和 ui.router，还有一种为 Angular2 设计的 new router（面向组件）。后面那个没在实际项目中用过，就不讲了。

无论是 ngRoute 还是 ui.router，作为框架额外的附加功能，都必须以 模块依赖 的形式被引入。

区别

ngRoute 模块是 Angular 自带的路由模块，而 ui.router 模块是基于 ngRoute模块开发的第三方模块。

ui.router 是基于 state （状态）的， ngRoute 是基于 url 的，ui.router模块具有更强大的功能，主要体现在视图的嵌套方面。

使用 ui.router 能够定义有明确父子关系的路由，并通过 ui-view 指令将子路由模版插入到父路由模板的 <div ui-view></div> 中去，从而实现视图嵌套。而在 ngRoute 中不能这样定义，如果同时在父子视图中 使用了 <div ng-view></div> 会陷入死循环。

分属不同团队进行开发的 angular 应用，如果要做整合，可能会遇到哪些问题，如何解决？

可能会遇到不同模块之间的冲突。

比如一个团队所有的开发在 moduleA 下进行，另一团队开发的代码在 moduleB 下

angular.module('myApp.moduleA', [])

.factory('serviceA', function(){

...

})

angular.module('myApp.moduleB', [])

.factory('serviceA', function(){

...

})

angular.module('myApp', ['myApp.moduleA', 'myApp.moduleB'])

会导致两个 module 下面的 serviceA 发生了覆盖。

貌似在 Angular1.x 中并没有很好的解决办法，所以最好在前期进行统一规划，做好约定，严格按照约定开发，每个开发人员只写特定区块代码。

angular 的缺点有哪些？

强约束

导致学习成本较高，对前端不友好。

但遵守 AngularJS 的约定时，生产力会很高，对 Java 程序员友好。

不利于 SEO

因为所有内容都是动态获取并渲染生成的，搜索引擎没法爬取。

一种解决办法是，对于正常用户的访问，服务器响应 AngularJS 应用的内容；对于搜索引擎的访问，则响应专门针对 SEO 的HTML页面。

性能问题

作为 MVVM 框架，因为实现了数据的双向绑定，对于大数组、复杂对象会存在性能问题。

可以用来 优化 Angular 应用的性能 的办法：

减少监控项（比如对不会变化的数据采用单向绑定）

主动设置索引（指定 track by，简单类型默认用自身当索引，对象默认使用 $$hashKey，比如改为 track by item.id）

降低渲染数据量（比如分页，或者每次取一小部分数据，根据需要再取）

数据扁平化（比如对于树状结构，使用扁平化结构，构建一个 map 和树状数据，对树操作时，由于跟扁平数据同一引用，树状数据变更会同步到原始的扁平数据）

另外，对于Angular1.x ，存在 脏检查 和 模块机制 的问题。

移动端

可尝试 Ionic，但并不完善。

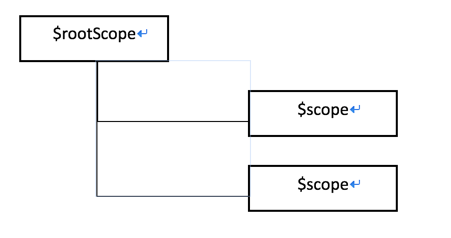
详述 angular 的 “依赖注入”？

栗子

依赖注入是一种软件设计模式，目的是处理代码之间的依赖关系，减少组件间的耦合。

参考：<https://segmentfault.com/a/1190000005836443>

解释下什么是$rootScrope以及和$scope的区别？

通俗的说$rootScrope 页面所有$scope的父亲。

如何产生$rootScope和$scope吧。

step1:Angular解析ng-app然后在内存中创建$rootScope。

step2:angular回继续解析，找到{{}}表达式，并解析成变量。

step3:接着会解析带有ng-controller的div然后指向到某个controller函数。这个时候在这个controller函数变成一个$scope对象实例。

如何取消 $timeout, 以及停止一个$watch()?

停止 $timeout我们可以用cancel：

var customTimeout = $timeout(function () {

// your code

}, 1000);

$timeout.cancel(customTimeout);

停掉一个$watch：

// .$watch() 会返回一个停止注册的函数

function that we store to a variable

var deregisterWatchFn = $rootScope.$watch(‘someGloballyAvailableProperty', function (newVal) {

if (newVal) {

// we invoke that deregistration function, to disable the watch

deregisterWatchFn();

...

}

});

Angular Directive中restrict 中分别可以怎样设置？scope中@,=,&有什么区别？

restrict中可以分别设置:

A匹配属性

E匹配标签

C匹配class

M 匹配注释

当然你可以设置多个值比如AEC,进行多个匹配。

在scope中，@,=,&在进行值绑定时分别表示

@获取一个设置的字符串，它可以自己设置的也可以使用{{yourModel}}进行绑定的;

= 双向绑定，绑定scope上的一些属性；

& 用于执行父级scope上的一些表达式，常见我们设置一些需要执行的函数

]

列出至少三种实现不同模块之间通信方式？

1、Service

2、events,指定绑定的事件

3、使用 $rootScope

4、controller之间直接使用$parent, $$childHead等

5、directive 指定属性进行数据绑定

$apply()和 $digest()的区别

安全性：$apply()可以接收一个参数作为function()，这个 function 会被包装到一个 try … catch 块中，所以一旦有异常发生，该异常会被 $exceptionHandler service 处理。

$apply会使ng进入 $digest cycle , 并从$rootScope开始遍历(深度优先)检查数据变更。

$digest仅会检查该scope和它的子scope，当你确定当前操作仅影响它们时，用$digest可以稍微提升性能。

compile和link的区别： 看到一个比较6的答案。性能力（性能和能力）

编译的时候，compile转换dom，碰到绑定监听器的地方就先存着，有几个存几个，到最后汇总成一个link函数，一并执行，提升了性能。

function compile(tElement, tAttrs, transclude) { ... }tElement为编译前的element

function link(scope, iElement, iAttrs, controller) { ... } iElement为编译后的element，已经与作用域关联起来，所以可以数据绑定

；如果指令只进行DOM的修改，不进行数据绑定，那么配置在compile函数中，如果指令要进行数据绑定，那么配置在link函数中。