**一、选择题**

1. 下面哪些不属于客户端存储机制（）

A.cookie B.localstorage C.appcache D.session E.sessionstorage

2.TCP和IP分别对应了OSI中的哪几层（）

A.Application layer B.Presentation layer

C.Transport layer D.Network layer

3.浏览器访问某页面，HTTP协议返回状态码为403时表示：（）

A.找不到该页面 B.禁止访问

C.内部服务器访问 D.服务器繁忙

4.下面几种点击页面的按钮，使之加载一个网页，哪几种可以实现在新窗口打开（）

A.<input type=”button” value=”new” onclick=”open(‘new.html’,’\_blank’)”/>

B.<input type=”button” value=”new” onclick=”window.location=’new.html’;”/>

C.<input type=”button” value=”new” onclick=”location.assign(‘new.html’);”/>

D.<form target=”\_blank” action=”new.html”><input type=”submit” value=”new”/></form>

5.下面语句中哪些可以正确的创建一个数组（）

A．var array=new Array();

B. var array={‘’,a,b,1};

C. var array=[‘’,’1’,’a’,4];

D. var array=(‘’,1,2,3);

6.下面程序的result值是（）

function justTest(num){

if(num>3){

return num+1;

}

}

var result=justTest(2);

A．3 B.程序出错 C.Null D.4 E.Undefined

7.foo对象有att属性，怎么获取att的值：（）

A.foo.att B.foo{“att”}

C.foo.getAttribute(“att”) D.foo(“att”)

8.下面程序的result值是（）

A．258 B.”258” C.78 D.15 E.”78”

1. **解答题**

1.用到localstorage的什么方法？

设置：setItem()

读取：getItem()

修改（即重新设置）：setItem()

删除某一个：removeItem()

删除全部：clear()

1. 怎么用cookie的，做了什么？

cookie的适用范围：  
 1:保存上次登录时间等信息   
 2:保存用户名、密码，在一定时间不用重新登录  
 3: 记录用户访问网站的喜好(比如有无背景音乐、网页的背景色是什么)  
 4: 网站的个性化，比如定制网站的服务，内容。  
 Session的使用范围：  
 1：网上商城中的购物车  
 2：保存登录用户的信息  
 3：将某些数据放入到Session中，供同一用户的各个页面使用  
 4: 防止用户非法登录  
 Session和Cookie的主要区别在于：  
 1: Cookie是把用户的数据写给用户的浏览器。  
 2: Cookie可以被用户禁用，但是sessio用户的数据写到用户独占的$\_SESSION中,存在服务器的某个路径的文件中。

1. Cookie的过期和Session的超时有什么区别？

会话的超时由服务器来维护，它不同于Cookie的失效日期。首先，会话一般基于驻留内存的cookie，不是持续性的cookie，因而也就没有截至日期。即使截取到JSESSIONID cookie，并为它设定一个失效日期发送出去。浏览器会话和服务器会话也会截然不同。

1. splice和delete有什么区别？

splice删除数组，改变了原数组，数组的长度变化了

delete删除数组，只是被删除的元素变成了undefined，其他的元素的键值还是不变

1. 怎么做分页？以及分页页码？

案例里有写，结合react课上案例

1. ajax获取数据的整个流程？

## 一、Ajax的对象 XMLHttpRequest的方法

| **方法** | **描述** |
| --- | --- |
| getAllResponseHeaders() | 把 HTTP请求的所有响应首部作为健／值对返回 |
| getResponseHdader(“header”) | 返回指定首部的串值 |
| open(“method”,”url”) | 建立对服务器端调用。Method参数可以是GET、POST或PUT。  url参数可以是相对URL或绝对URL |

## 二、发送请求

### ****setRequestHeader(header,value)****

当浏览器向服务器请求页面时，它会伴随着这个请求发送一组首部信息。这些首部信息是一系列描述请求的元数据（metadata）.首部信息用来声明一个请求是GET还是 POST。

Ajax请求中，发送首部信息的工作可以由setRequestHeader完成参数header :首部的名字；参数 value：首部的值如果用POST请求向服务器发送数据，需要将”Content-type” 的首部设置为 “application/x-www-form-urlencoded”.它会告知服务器正在发送数据，并且数据已经符合URL编码来。该方法必须在open()之后才能调用

## 三、接收响应

### ****readyState****

readyState属性表示Ajax请求的当前状态。它的值用数字代表。

* 0 代表未初始化，还没有调用open 方法
* 1 代表正在加载，open方法已经被调用，但 send方法还没有被调用
* 2 代表加载完毕。send方法已被调用，请求已经开始
* 3 代表交互中。服务器正在发送响应
* 4 代表完成。响应发送完毕

每次readyState值的改变，都会触发readystatechange事件。如果把onreadystatechange事件处理函数赋给一个函数，那么每次 readyState的值的改变都会引发该函数的执行。   
readyState值的变化会因浏览器的不同而有所差异。但是，当请求结束的时候，每个浏览器都会把readyState的值统一设为4.

实例代码呈现

<!doctype html><html lang="en"><head>

<meta charset="UTF-8">

<title>Document</title>

<script>

window.onload=function(){

//1、获取a节点，并为其添加onclick响应函数

document.getElementsByTagName("a")[0].onclick=function(){

//3、创建一个XMLHttpRequest对象

var request=null;

request= new XMLHttpRequest();

//4、准备发送请求对数据：url

//增加时间戳起到禁用缓存对目的

var url= this.href+"?time="+new Date();

//var method="GET";发送数据的方法

var method="POST";

//5、调用XMLHttpRequest 对象对open方法

request.open(method,url);

request.setRequestHeader("ContentType","application/x-www-form-urlencoded");

//6、调用XMLHttpRequest对send方法

request.send("name='atguigu'");

//7、为XMLHttpRequest 对象添加onreadystatechange响应函数

request.onreadystatechange=function(){

if(request.readyState==4){

if(request.status==200||request.status==304){

alert(request.responseText);

}

}

}

//8、判断响应是否完成： XMLHttpRequest对象对readyState 属性值为4对时候

//9、再判断响应是否可用：XMLHttpRequest对象对status属性值为200

//10、打印响应结果：responseText

//2、取消a节点对默认行为

return false;

}

}

</script></head><body>

<a href="helloAjax.txt" >Ajax</a></body></html>

7. 状态码200，404，405，505分别代表什么？

1xx - 信息提示

2xx - 成功

3xx - 重定向

4xx - 客户端错误

5xx - 服务器错误

200：OK 一切正常，对GET和POST请求的应答文档跟在后面。

404：Not Found 无法找到指定位置的资源。

405：Method Not Allowed 请求方法（GET、POST、 HEAD、Delete、PUT、TRACE等）对指定的资源不适用，用来访问本页面的 HTTP 谓词不被允许（方法不被允许）（HTTP 1.1新）

505：HTTP Version Not Supported 服务器不支持请求中所指明的HTTP版本。（HTTP 1.1新）。

8. H5的特点？

1.可以进行跨平台的使用；

　 2、提高可用性和改进用户的友好体验；

　　3、有几个新的标签，这将有助于开发人员定义重要的内容；

　　4、可以给站点带来更多的多媒体元素(视频和音频)；

　　5、可以很好的替代FLASH和Silverlight；

　　6、当涉及到网站的抓取和索引的时候，对于SEO很友好；

　　7、将被大量应用于移动应用程序和游戏。

9. 介绍一下原型链？

每一个函数都有一个属性叫做prototype，prototype的属性值是一个对象，默认的只有一个叫做constructor的属性，指向这个函数本身。每一个对象都有一个隐藏的属性\_\_proto\_\_，这个属性引用了创建这个对象的函数prototype。访问一个对象的属性时，先在基本属性中查找如果没有，则会沿着\_\_proto\_\_这条链向上找，这就是原型链。

10.原生的javascript获取指定元素下所有的元素节点

11.写出一些前端优化的策略

**前端优化是什么**：从用户访问资源到资源完整的展现在用户面前的过程中，通过技术手段和优化策略，缩短每个步骤的处理时间从而提升整个资源的访问和呈现速度。

**为什么要去做优化**：在构建web站点的过程中，任何一个细节都有可能影响网站的访问速度，如果不了解性能优化知识，很多不利网站访问速度的因素会形成累加，从而严重影响网站的性能，导致网站访问速度变慢，用户体验低下，最终导致用户流失。

**从浏览器发起请求到页面能正常浏览需要经历的阶段有**：预处理——>DNS解析——>建立连接——>发起请求——>等待响应——>接受数据——>处理元素——>布局渲染

**优化策略**：

一）内容层面

1、DNS解析优化（DNS缓存、减少DNS查找、keep-alive、适当的主机域名）

2、避免重定向（/还是需要的）

  3、切分到多个域名

  4、杜绝404

二）网络传输阶段

1、减少传输过程中实体的大小

    1）缓存

    2）cookie优化

    3）文件压缩（Accept-Encoding：g-zip）

2、减少请求的次数

    1）文件适当的合并

    2）雪碧图

3、异步加载（并发,requirejs）

4、预加载、延后加载、按需加载

三）渲染阶段

1、js放底部，css放顶部

2、减少重绘和回流

  3、合理使用Viewport 等meta头部

  4、减少dom节点

  5、BigPipe

四）脚本执行阶段

1、缓存节点，尽量减少节点的查找

2、减少节点的操作（innerHTML）

3、避免无谓的循环，break、continue、return的适当使用

4、事件委托

12.写出用到的移动端的适配方案

( <http://www.jianshu.com/p/e5ca5b78e03e> 里面有各自方案的详解及优缺点)

百分比适配

通过媒体查询的方式即CSS3的meida queries

以天猫首页为代表的 flex 弹性布局

以淘宝首页为代表的 rem+viewport缩放

rem 方式

13.svn和git的区别

<http://www.jianshu.com/p/bfec042349ca>

五个基本区别：

1. GIT是分布式的，SVN不是；
2. GIT把内容按元数据方式存储，而SVN是按文件；
3. GIT分支和SVN的分支不同；
4. GIT没有一个全局的版本号，而SVN有；
5. GIT的内容完整性要优于SVN。

14.有没有用过Nunjucks

<https://www.liaoxuefeng.com/wiki/001434446689867b27157e896e74d51a89c25cc8b43bdb3000/0014713964925087c29166d8c344a949364e40e2f28dc09000>

Nunjucks是Mozilla开发的一个纯JavaScript编写的模板引擎，既可以用在Node环境下，又可以运行在浏览器端。但是，主要还是运行在Node环境下，因为浏览器端有更好的模板解决方案，例如MVVM框架。

15.详细介绍一下同步异步的原理

JS单线程：

所谓单线程，是指在JS引擎中负责解释和执行的JavaScript代码的线程只有一个。我们叫它主线程。

但是实际上还存在其他的线程。例如：处理ajax请求的线程、处理DOM事件的线程、定时器线程、读写文件的线程（例如在Nodejs中）等等。这些线程可能存在于JS引擎之内，也可能存在于JS引擎之外，我们把它们叫做工作线程。

同步与异步：

同步：提交后等待服务器的响应，接收服务器返回的数据后再执行下面的代码

异步：提交后继续执行下面的代码，而在后台继续监听，服务器响应后有程序做相应处理。

异步操作的好处是不必等待服务器而可以继续在客户端做其他事情

常见的异步操作有：

回调函数：

A(args...,callbackFn)

发起函数（或叫注册函数）A

回调函数callbackFn,其中回调函数要等前面args...执行完之后再开始

Ajax请求

定时器setTimeout

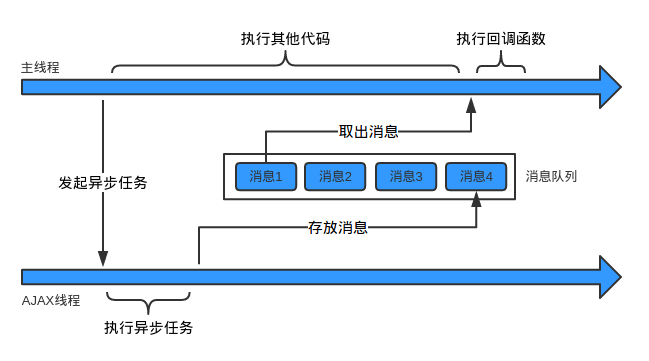
消息队列与事件循环：

在异步过程中，工作线程在异步操作完成后利用消息队列和事件循环通知主线程。（工作线程将消息放到消息队列，主线程通过事件循环过程去取消息）

消息队列：消息队列是一个先进先出的队列，它里面存放着各种消息。

事件循环：事件循环是指主线程重复从消息队列中取消息、执行的过程。

实际上，主线程只会做一件事情，就是从消息队列里面取消息、执行消息，再取消息，再执行。当消息队列为空时，就会等待直到消息队列变成非空。而且主线程只有在当前的消息执行完成后，才会去取下一个消息。这种机制叫做**事件循环机制**，取一个消息并执行的过程叫做一次循环。



异步过程的回调函数，一定不在当前这一轮事件循环中执行。

16.jquery中的hasClass用原生怎么完成

attributes[class].val获取类----->split(‘ ’)字符串切割----->遍历数组一一去比较

17.rem的算法

一般移动端设计稿为640\*1136,（px）,假设我们设置html的font-size为10px即1rem=10px;我们以iphone5为浏览设备为标准制作，那么我们在切图写页面的时候，rem和px这个关系是怎样处理的呢？（注：iphone5分辨率为320\*568px）  
答：假定我们从设计稿上切一张480px的图，我们的rem计算过程为：  
320/640=1/2； //应该设置的尺寸占设计稿尺寸多少比例  
480\*（1/2）/10=480/20=24rem  
总结公式：设计稿任意尺寸px\*（设备屏幕px/设计稿尺寸px）/(font-size)；

18.兼容性问题（H5，css，js）？IE8以上的兼容有哪些？hack？

(<http://blog.csdn.net/swallowliyan/article/details/48137453>)

19.$(this)和this关键字在jQuery中有何不同？

$(this)返回一个jQuery对象，你可以对它调用多个jQuery方法。

this代表当前元素，它是JavaScript关键词中的一个，表示上下文中的当前DOM元素，你不能对它调用jQuery方法，直到它被$()函数包裹，例如$(this)。

20.写出如下表达式的结果

NaN == NaN //false

typeof(NaN) //number

typeof(null) //object

typeof(undefined)//undefined

false || true //true

false && true //false

2 && 3 //3

2 || 3 //2

21.写出window.onload与$(function(){})的区别

Window.onload等待页面全部加载完毕再执行下面的代码，包括图片、文字等等；$(function(){})等待页面的结构加载完毕就执行下面的代码，不需要等图片加载完毕。

22.请用三种方式写出4平分一个div的布局方式（至少是一种flex布局）

方法一：父盒子加display:flex;

方法二：采用bootstrap，子盒子加class="col-sm-3"；

方法三：column-count:4;必须设置父盒子的宽；

方法四：浮动

23.为什么利用多个域名来存储网络资源会更有效

1. 更方便CDN缓存

将静态内容和动态请求分别存放在不同服务器上，更加方便进行CDN缓存

1. 突破浏览器并发限制

每个浏览器都具有一个并发浏览请求数，表示的是同一时刻对于同一域名下的请求数有一定的限制，超出这个限制后的请求将不被接受，直到等到有空闲连接数。

1. 节省cookie带宽

因为在很多时候，对于页面资源的请求我们是不需要携带用户cookie的，比如浏览图片，视频等信息，而用户登录时才需要用户cookie信息。此时，就可以选择将图片，视频等资源放在专门存放静态资源的域名下，节省发送请求时，传递cookie需要的带宽。

1. 节省主域名连接数

一般用户在访问网站的时候会访问主域名对应的网页，根据之前提到的浏览器并发连接数，对应主域名页面的访问连接就显得尤为珍贵。如果将不同资源放在不同域名下，则可以节省主域名连接数，从而能优化页面的访问速度。

注意：并非同一个网站资源在越多的域名下就越好，原因有两点：

1. 因为浏览器在做DNS解析时也会花费时间，太多的域名会导致解析时间过长，访问速度变慢。
2. 如果域名采用的是HTTPS协议，则需要安装更多的证书，在部署时也存在一定的问题，加大了操作难度。

24.box-sizing常用的属性有哪些？分别有什么作用？

content-box:宽度和高度分别应用到元素的内容框，在宽度和高度之外会之元素的内边距和边框；

border-box：为元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度；

inherit:规定应从父元素继承box-sizing属性的值。

25.call()和apply()的区别和作用

call()和apply()的功能是一样的 ，都可以用来代替另一个对象调用一个方法，将一个函数的对象上下文从初始的上下文改变为由thisObj指定的新对象。

区别就在于写法不同：

call方法：B.call(A,args1,args2)

apply方法：B.apply(A,args1,args2)

用法：

1. 调用另一个对象的方法；
2. 改变this指向；
3. 实现继承；
4. 多重继承。

26.什么是跨站请求伪造？以及如何解决？

用户A 用浏览器 访问了博客园网站，登录后，会把登录信息保存在用户A的浏览器cookie中。这时用户A访问了攻击人的网站 B,但是B网站一加载到用户A的浏览器中，就执行一段js文件，这个js文件是向博客园网站请求改用户密码，那么博客园网站接到这个请求时，会验证，由于之前用户A的浏览器cookie中保存了登录信息，博客园网站会认为这是用户的操作，就直接执行。这就是所谓的请求伪造。

**解决方法：**

可以通过服务器端的随机数token（最安全）判断

 1、Token的引入

Token是在客户端频繁向服务端请求数据，服务端频繁的去数据库查询用户名和密码并进行对比，判断用户名和密码正确与否，并作出相应提示，在这样的背景下，Token便应运而生。

2、Token的定义

Token是服务端生成的一串字符串，以作客户端进行请求的一个令牌，当第一次登录后，服务器生成一个Token便将此Token返回给客户端，以后客户端只需带上这个Token前来请求数据即可，无需再次带上用户名和密码。

3、使用Token的目的

Token的目的是为了减轻服务器的压力，减少频繁的查询数据库，使服务器更加健壮。

referer解决。

HTTP Referer是header的一部分，当浏览器向web[服务器](https://baike.so.com/doc/4487696-4696885.html)发送请求的时候，一般会带上Referer，告诉服务器我是从哪个页面链接过来的，服务器籍此可以获得一些信息用于处理。

27.

<ul id=”list” class=”foo”>

<li>#0</li>

<li><span>#1</span></li>

<li>#2</li>

<li>#3</li>

<li><ul><li>#4</li></ul></li>

...

<li><a href=”//v2ex.com”>#99998</a></li>

<li>#999999</li>

<li>#100000</li>

</ul>

为<ul>添加一个类bar

删除第10个<li>

在第500个<li>后面增加一个<li>，其文字内容为<yxfin.com/>

点击任意<li>弹窗显示其为当前列表的第几项

要求（1.用es6写；2.请多考虑dom变化和性能）

$('ul').addClass('bar')

$('#list').find('li').eq(9).remove()

$('#list').find('li').eq(500).append(function(){

return "<li><yxfin.com/>aaaaaaaa</li>"

})

$('#list').on('click','li',function(){

console.log($(this).index())

})

28.下列哪一项会返回”true”

null==undefined //true

Boolean(“true”) //true

Boolean(0) //false

isNaN(10) //false

isNaN(“10”) //false

NaN === NaN //false

29.请描述一下cookies,sessionStorage和localStorage的区别？

**共同点：**

都是保存在浏览器端，且同源的。

**区别：**

cookie数据始终在同源的http请求中携带（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。cookie数据还有路径（path）的概念，可以限制cookie只属于某个路径下。

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。

30.什么是“use strict”？使用它的好处和坏处分别是什么？

use strict -----------严格模式，使得js在更严格的条件下运行

优点：

1. 消除js语法的一些不合理、不严谨之处，减少一些怪异行为；
2. 消除代码运行的一些不安全之处，保证代码运行的安全；
3. 提高编译器效率，增加运行速度；
4. 为未来新版本的JS做好铺垫；

注：经过测试IE6,7,8,9均不支持严格模式

缺点：

现在网站的js都会进行压缩，一些文件用了严格模式，而另一些没有。这些本来是严格模式的文件，被merge后，这个串就到了文件的中间，不仅没有指示严格模式，反而在压缩后浪费了字节。

31.简述JQuery中bind/live/on的区别？如何实现自定义事件？

Bind是用来绑定一个或多个事件的，live其实是bind的一个加强版，用来绑定通过JavaScript或者jQuery添加的DOM元素事件，其语法结构是一样的。Delegate用于事件委托，on是在1.7之后新增的，是把bind，live，delegate全部合并在一起。

32.restrict的值可以是哪些？

E 只限元素名使用

C 只限类名使用

M 只限注释使用

A 只限属性使用

restrict默认值为EA，即可以通过元素名和属性名来调用指令。

33.目前JS单元测试框架你知道哪些？

[QUnit](https://github.com/jquery/qunit)、[Jasmine](http://pivotal.github.com/jasmine/" \t "https://q.cnblogs.com/q/32635/_blank)、[Jarvis](http://jarvis.tmont.com/" \t "https://q.cnblogs.com/q/32635/_blank)、[jfUnit](http://felipenmoura.org/projetos/jfunit/home.php" \t "https://q.cnblogs.com/q/32635/_blank) 等，推荐[QUnit](https://github.com/jquery/qunit" \t "https://q.cnblogs.com/q/32635/_blank)。（具体的有文档可以去研究一下<http://qunitjs.com/>）

1. 跨域的方法有哪些？jsonp的原理是什么？如果页面编码和被请求的资源编码不一致该如何处理？

**跨域方法：**

1. 通过document.domain进行不同子域框架间的交互；
2. 通过location.hash跨域；
3. 通过HTML5的postMessage方法跨域；
4. 通过jsonp跨域；
5. 通过cors跨域；
6. 通过window.name跨域

**jsonp原理：**

通过script标签引入的js是不受同源策略限制的，故而通过script标签引入一个js文件此文件返回一个js函数的调用。

**编码不一致处理办法：**

<script src="<http://www.xxx.com/test.j>s" charset="utf-8"></script>

在引入的时候，了解对方的编码格式，加上对方的编码

1. 写出你对this的理解

在函数中this到底取何值， 是在函数真正被调用执行的时候确定的，函数定义的时候确定不了。因为this的取值是执行上下文环境的一部分，每次调用函数，都会产生一个新的执行上下文环境。谁去调用，则this指代谁。

1. 实现javascript异步编程的方法有哪些

回调函数、订阅与发布、promise、事件监听

1. js中call，apply，callee，caller用法及含义

<http://www.cnblogs.com/yuzhongwusan/archive/2012/02/13/2348782.html>