# http中get和post的区别

get用于获取数据，post用户修改数据

get提交数据基于url，不同浏览器对url的长度有不同的限制

get提交数据速度更快

post没有对数据的大小限制

post更加安全，get提交的数据，可能会被缓存

# jquery绑定的事件和命名空间

任何名称均可作为绑定的事件名称，标准事件名称可通过用户操作执行回调方法，非标准和标准事件名称均可通过.tigger()来触发。

命名空间指click.a.b，其中a，b就是命名空间，用于相同dom元素绑定多个同名事件，再指定触发某个事件和指定删除某个事件时使用

# 数组去重

var *array* =['a','b','1','c','a',1,36];  
var distinct = function(array){  
 var arr = [];  
 for(var i = 0; i<array.length; i++){  
 arr.indexOf(array[i])< 0 && arr.push(array[i])  
 }  
 return arr  
}  
distinct(*array*);

# display的值

block,inline,inline-block,none

none:不被显示

block:块级元素，前后形成断行

Inline：内联元素，前后没有断行，长度跟内容相关

Inline-block:行内块元素，前后没有断行，但是可以像block一样，设置宽度高度等

# 5.mouseover 和mouseenter 的区别

不论鼠标穿过被选元素还是其子元素都会触发mouseover对应mouseout

只有在鼠标穿过被选元素时，才会触发mouseenter对应mouseleave

# 6.javascript有几种数据类型

在Javascript中只有五种简单类型，分别为null,undefined,boolean,String和Number。一种复杂类型：object。代码类型只有一种形式就是function。

# domReady和onload的区别

*/\* 在document文档加载完成后执行，document文档包括了加载图片等其他信息\*/  
/\*\*  
 \* 多个window.onload 仅执行一个，  
 \*/*window.onload = function(){  
 console.log('onLoad2')  
};  
*/\*\*  
 \* 绑定方式的load 可执行多个  
 \*/*window.addEventListener( "load", function(){  
 console.log(new Date(),'load1')  
});  
window.addEventListener( "load", function(){  
 console.log(new Date(),'load2')  
});  
  
*/\*在DOM加载完成后执行，在执行同步js代码之后，渲染图片等其他信息之前\*/*document.addEventListener('DOMContentLoaded',function(){  
 console.log($('img').width())  
 console.log(new Date(),'DOMContentLoaded')  
},false);

# 8.http状态码

200 ：OK  
301：永久重定向  
302：临时重定向

304：文件在服务器上未被修改，可以使用缓存  
404：不存在  
500：服务器出错

# 9.单例模式

var *Singleton* = (function () {  
 var instantiated;  
 function init() {  
 */\*这里定义单例代码\*/* return {  
 publicMethod: function () {  
 console.log('hello world');  
 },  
 publicProperty: 'test'  
 };  
 }  
  
 return {  
 getInstance: function () {  
 if (!instantiated) {  
 instantiated = init();  
 }  
 return instantiated;  
 }  
 };  
})();

# 深度克隆一个对象

function cloneObj(obj){  
 var result, i,item;  
 if(result === undefined){  
 if(Object.prototype.toString.call(obj) === '[object Array]'){  
 result = [];  
 for(i = 0; i<obj.length; i++){  
 result[i] = cloneObj(obj[i]);  
 }  
 }else if(Object.prototype.toString.call(obj) === '[object Object]'){  
 result = {};  
 for(item in obj){  
 result[item] = cloneObj(obj[item]);  
 }  
 }else{  
 result = obj;  
 }  
 }  
 return result;  
}

# 请简述javascript延迟加载的方式

# Arguments

永远与命名参数的值相同，修改arguments也会修改命名参数的值，修改命名参数的值，也会修改arguments的值，arguments的长度是由传入参数的个数决定的，不是由函数定义时的命名参数个数决定的。如果只传入了一个参数，那么arguments[1]设置的值不会反应到命名参数中。

arguments.callee ：callee属性是arguments 对象的一个成员，它表示对函数对象本身的引用.

arguments.callee.length ：代表函数的形参长度  
arguments.length:代表实参长度

function.length ：代表形参长度

caller：function.caller 属性只有在函数执行时才有定义。caller这个属性保存哪个函数调用了当前方法

Array.prototype.slice.call(likeArray) 将类数组转换为数组

# 13.一个页面从输入 URL 到页面加载显示完成，这个过程中都发生了什么？

webKit 且不考虑缓存的情况：

1. 通过http请求将html页面下载至浏览器中
2. 将html代码自上到下依次解析
3. 找到需要下载的js(一些浏览器不是并行下载，按先后顺序阻塞下载)，css，img等外部元素，并行下载
4. 页面加载流程跟css文件链接在html中的位置无关，跟js文件在html代码中的位置有关
5. 所有Css文件下载完成前，浏览器不会渲染dom
6. Js文件下载完成，但css文件下载未完成，不渲染dom，js代码也不会执行
7. 所有css文件下载完成后，在html代码中依次从上至下渲染dom
8. 碰到script标签，若js文件未下载完成，因浏览器无法判断js代码是否会对渲染造成印象，会等待js下载完成，阻塞UI渲染，等待js下载完成后，执行js代码，执行代码后，继续向下渲染dom
9. Dom渲染完成后，触发通过js绑定的DOMContentLoad事件
10. Image下载完成并渲染后，触发通过js绑定的onload 事件

# typeof

数字：number

字符串：string

布尔：boolean

方法：function

undefined:undefined

未定义:undefined

null:object

其他：object

# + - \* / == ===

**+:** 1.两边都是数字或字符串时，正常处理

2.一边包含字符串，另一边强转成字符串，做字符串拼接操作

3.两边均为非字符串的原始值(null,undefined,number,boolean),强制转换为数字，做数字 加操作

4.左边存在引用值，则将两边均转换为字符串，做字符串拼接操作

总结:相加时，存在字符串或对象，按字符串操作，否则按数字操作

**==：**1.一个null，一个undefined，则相等

1. 两个都是数字，两个都是字符串，正常比较
2. 一个数字，一个字符串，将字符串转换为数字后比较

2.3.true转换为1，false转换为0，转换后再比较

4.两个对象，比较两个对象的引用值

5.一个对象，一个字符串或数字，将对象转换为字符串后比较

# 自定义异常

var myError = function(message){  
 this.message = message;

}

myError.prototype = new Error();

抛出异常:throw myError(‘自定义异常’);

判断自定义异常：e instanceof myError

# 高性能网站建设

1.图片地图？

2.CSS Sprites 将多张图片合并为一张图片，通过background-position 指定显示的内容

3.内联图片，将图片转为base64的格式，整体下载量会增加

4.合并js文件，合并css文件

5.使用CDN，用户离服务器越近，http请求越快

6.使用缓存cache-control max-Age

7.样式表放在最顶部，放在顶部先显示页面，再显示图片，放在底部，等待图片全部下载完成，整体显示。放在底部一些情况下，会出现页面闪烁

8.脚本放到最底部

9.并行下载：HTTP1.1规范里，建议浏览器从每个主机名并行下载两个组件，不同浏览器不同，但对于同名主机，并行下载量是固定的，当数量过多，必然出现排队现象

10.压缩js，css代码

11.304状态码，检测浏览器缓存的文件与服务器上的文件是否匹配

1,比较最新修改时间(Last-Modified,If-Modified-Since)

2,比较标签（ETag）集群换将，会出现相同文件，不同服务器返回的ETag不同

标签比较的优先级大于时间比较

# 高性能javascript

测试js文件并行下载（多浏览器）**defer**

1. 将脚本放在页面底部
2. 打包js文件
3. 延期下载脚本，
4. 指定script标间的defer属性，dom加载完成前，不会执行
5. 动态脚本元素,多文件的执行顺序

var *script* = document.createElement ("script");  
*script*.type = "text/javascript";  
*script*.src = "file.js";  
*script*.onload = function(){  
 alert('script success')  
};  
*script*.onreadystatechange = function(){  
 if (*script*.readyState == "loaded" || *script*.readyState == "complete"){  
 *script*.onreadystatechange = null;  
 alert("Script loaded.");  
 }  
};  
document.getElementsByTagName("head")[0].appendChild(*script*);

1. XHR脚本注入，通过ajax方式获取js文件内容，动态插入到script内，主要缺点是 文件和页面必须在同一域内
2. 作用域：每个函数都会有一个内部属性（Scope），Scope中包含作用域中对象的集合，称为作用域链，其中每个对象为可变对象，均有键值对组成，当一个函数被创建，它的作用域链被填充以对象，定义好此函数可以访问的外部数据。

运行函数时，创建一个运行时上下文，每次运行，创建一个，结束则销毁，创建运行时上下文时，会创建一个运行时作用域链，将函数的Scope复制到运行时作用域链中，将局部变量推入运行时作用域链的前端。

更改作用域链：

with，将with的内容推入运行时作用域链的最前端

try-catch

1. 闭包
2. 在函数中多次使用外部变量，最好将外部变量缓存起来，减少作用域链的查找
3. 在函数中多次使用对象的属性，最后将属性值缓存起来，减少原型链的查找，较少属性查找
4. 将多次的dom操作合并到一次执行

# Javascript权威指南

1.函数表达式可以包含名称，在递归中使用，其他情况下，会报错

var *obj* = {  
 f:function fact(x){  
 if(x>0){  
 return x  
 }else{  
 return fact(x + 1);  
 }  
 }  
}  
  
*obj*.f(-10)

1. 作用域链

每个函数在创建的时候都生成一个作用域链，该作用域链保存了函数内可以调用的外部变量，在函数执行时，创建一个新的对象来保存局部变量，并创建一个函数调用作用域链，将之前保存的作用域链推入底部，并将局部变量对象推入顶部

1. 闭包
2. Function 构造函数，new Function(‘x’,’y’,’return x + y;’)最后一个参数是函数体，之前的参数是函数的实参，通过构造函数创建的函数，无法捕获局部作用域，只能调用全局变量
3. 正则表达式

分组不捕获 (?: )

分组生成引用 \1 \2 \3 必须与分组匹配的值完全相同

正向前瞻 (?=exp)

负向前瞻 (?!exp)

String 方法：

search:返回与正则匹配的第一个字符，没有返回-1，忽略全局检索 string.search(reg)

Replace:第二个参数方法中的形参

Match:全局匹配，数组返回所有匹配的值，非全局匹配，索引0代表匹配的值，其 他索引代表正则中的分组捕获的值,没有匹配到返回null

split

正则对象

属性：

source:文本

global:是否是全局匹配

ignoreCase:是否忽略大小写

multiline:是否多行匹配

lastIndex: exec 和 test 用到

方法：

exec:与string的match方法类似，返回匹配的数组，不匹配返回null，但无论 是否为全局检索，数组均返回一个匹配的值，其他索引为分组捕获的值

当正则为全局匹配时，通过exec方法获取到返回值，会将正则的lastIndex 进行修改，下次匹配时，会匹配下一个匹配的值，直到返回null，将lastIndex 修改为0

test:判断字符转是否满足正则，当有全局匹配时，判断为true时，会修改 lastIndex，下次判断会从lastIndex 开始查找

1. 自动分号：不是在所有换行位置都添加分号，只有缺少分号无法正常解析时，才会添加分号；return，break，continue，后有换行，直接增加分号；
2. 数字操作：
   1. 无穷大的加减乘除还是无穷大，并保留正负号
   2. NaN !== NaN
   3. isNaN 参数为数字或数字字符串时返回false
   4. isFinite 参数为NaN Infinity 时返回true
   5. Infinity === Infinity
   6. Infinity !== -Infinity
3. undefined和null

undefined == null // true

Undefined === null // false

Number(undefined) // NaN

Number(null) // 0

1. 包装对象

var str = ‘qwe’;

str.length = 4;

console.log(str.length)

调用属性的时候，先生成一个对象，再销毁

原始值不可变更,修改原始值的变量，实际上是一个新的原始值

# 严格模式：

1.函数调用，this === undefined

2.无法修改arguments

3.不可使用arguments.callee

4.Call,apply 第一个参数不管是什么，都会变成this，非严格模式，null/undefined 都会变成window，其他原始值变为对应的包装对象

# Javascript精髓

Redonly和disable的区别

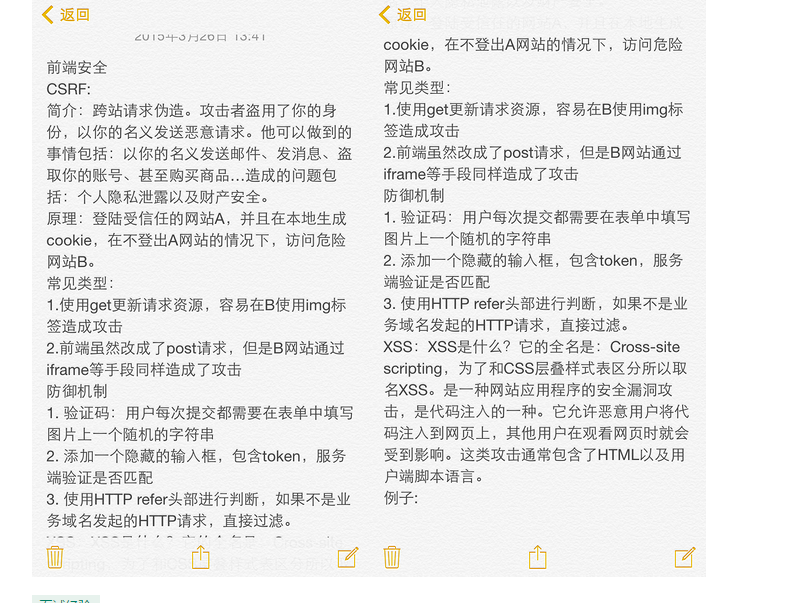
前端安全

浏览器内核

Seajs requirejs

圣杯布局

1. jQuery的四种绑定bind，live，delegate，on的区别？
2. 8.几个名词：ajax、XSS、session hijacking、content。
3. 9.闭包是什么？闭包的优缺点？闭包的使用场景？
4. 10.什么是冒泡事件？如何阻止冒泡事件？
5. Jquery api



11.Js面向对象的几种方式。

.jQuery中的$.ajax()方法的返回值具有什么特性

Define

闭包

1：对于用户体验有什么想法或见解  
  
1.取得一致性：  
类似的情况应该有让使用者有一致性的操作。在提示、选单与说明文件中，应该采用同样的名词。并且保持命令的一贯性。  
  
2.让重度用户使用快捷方式：  
当使用频率增加时，使用者会希望减少互动的次数、让每次的互动能够一次做更多的动作。缩写、功能键、隐藏功能与综观全局的功能，对专家来说非常有用。  
  
3.提供有意义的回馈：  
当使用者做出一些动作时，系统应该提供回馈。越频繁的动作，其回馈的强度可以低一些。越重要或不寻常的动作，其回馈强度应该要显著一些。  
  
4.设计对话产生结束：  
一连串的动作应该被组织成开始、中间、结束三部份。当动作结束的时候，要提供回馈让使用者知道动作已经完成。在做下个一连串的动作之前，先告知使用者整个流程，能够减轻使用者的压力、提高满意度。  
  
5.提供简单的错误处理：  
最好不要让系统有严重错误的可能性。如果还是造成错误，系统应该能够侦测出出来，并提供一个简单、使用者可以理解的错误处理方式。  
  
6.允许回到上一步：  
这个功能可以减低使用者的焦虑，因为使用者只到做错了可以重来。这个功能鼓励使用者探索不熟西的选项。回到上一步的功能，可以包含一个、或是一连串的动作。  
  
7.满足使用者控制的需求：  
有经验的使用者强烈的感觉到他们在控制系统，做出动作之后，系统提供回馈。系统设计上要让使用者作为动作的处发者，而不是响应者。  
  
8.减少短期记忆需求：  
人类的短期记忆有限，因此显示上要保持简单、能同时显示多页数据以减少窗口切换频率，减少记忆指令和动作顺序的时间。

1. css实现每个单词第一个字母大写。（这道题我不记得那css怎么写，不经常用，没写出来）。  
   2、overflow-x是css2还是css3。（我最不喜欢这种题，只要知道主流浏览器支不支持就足够了，记得前段时间在该论坛也说过这题，当时有人查是css2，有人查是css3。。。）  
   3、如果使用原生javascript深度克隆一个对象。  
   4、jquery中$(".class")和$("div.class")在IE8下哪个效率高，请解释原因。  
   5、以下哪个不是HTML5新标签：A、<article>  B、<section>  C、<address>  D、<time>  
   6、使用HTML和css实现效果（尽量不用float属性)

postMessage

引用传值，非引用传值

Websocket

自定义事件

内存泄漏

模板引擎

eval是做什么的？

null，undefined 的区别？

如何创建一个对象? （画出此对象的内存图）

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

JQuery的源码看过吗？能不能简单说一下它的实现原理？

Jquery Api

JQuery一个对象可以同时绑定多个事件，这是如何实现的？

WEB应用从服务器主动推送Data到客户端有那些方式？

请写一个 getParents 方法让它可以获取某一个 DOM 元素的所有父亲节点。

alert(Function instanceof Object);

alert(Object instanceof Function);

alert(Object instanceof Function); // return true 原因

alert(Function instanceof Object); // return true 原因

<http://www.w3cfuns.com/notes/15447/4991545ff0e5ec749345ac574f794619>

前端优化提高客户端响应速度的方法,至少10条以上

看过哪些书

严格模式与混杂模式——如何触发这两种模式，区分它们有何意义

DOM BOM