面试题4 js

**JavaScript**

1.用原生JavaScript的实现过什么功能吗？

2.介绍JavaScript的基本数据类型。

基本数据类型5种：字符串（string）、数字（number）、**布尔（**Boolean**）、空（Null）、未定义（Undefined）**

**复杂数据类型：数组、对象（跟本题没有多大关系，如果面试官换一种问法，就需要注意了，问**JavaScript数据类型都有哪些呢？**）**

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）如果你不知道数组的长度，使用 push

（3）请使用 ===/!== 来比较 true/false 或者数值

（4）对字符串使用单引号 ''(因为大多时候我们的字符串。特别html会出现")

（5）使用对象字面量替代 new Array 这种形式

（6）绝对不要在一个非函数块里声明一个函数，把那个函数赋给一个变量。浏览器允许你这么做，但是它们解析不同

（7）不要使用全局函数

（8）总是使用 var 来声明变量，如果不这么做将导致产生全局变量，我们要避免污染全局命名空间

（9）Switch 语句必须带有 default 分支

（10）使用 /\*\*...\*/ 进行多行注释，包括描述，指定类型以及参数值和返回值

（11）函数不应该有时候有返回值，有时候没有返回值

（12）语句结束一定要加分号

（13）for 循环必须使用大括号

（14）if 语句必须使用大括号

（15）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

（16）避免单个字符名，让你的变量名有描述意义

（17）当命名对象、函数和实例时使用驼峰命名规则

（18）给对象原型分配方法，而不是用一个新的对象覆盖原型，覆盖原型会使继承出现问题

（19）当给事件附加数据时，传入一个哈希而不是原始值，这可以让后面的贡献者加入更多数据到事件数据里，而不用找出并更新那个事件的事件处理器

参考文档：http://www.jb51.net/article/24579.htm

4.JavaScript原型(prototype)? 每个JS对象都有原型属性吗？

5.JavaScript有几种类型值？（堆：原始值和 栈：引用值），你能画一下他们的内存图吗？

栈：原始数据类型（Undefined，Null，Boolean，Number，String）

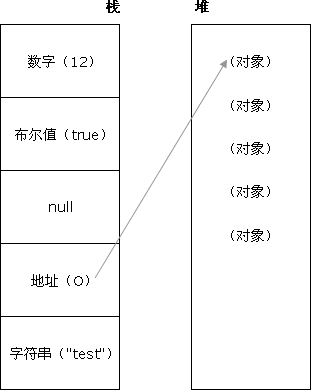
堆：引用数据类型（对象、数组、函数）

两种类型的区别：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 = 'song';

}

function Child() {

this.age = 28;

}

Child.prototype = new Parent(); //通过原型,继承了Parent

//

var demo = new Child()l;

alert(demo.age);

alert(demo.name); //得到被继承的属性

7.如何创建一个对象? （画出此对象的内存图）

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 = "Xiaosong";

person.age = "23";

person.work = function() {

alert("Hello " + person.name);

}

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 Person(name,age,hobby) {

this.name = name; //this作用域：当前对象

this.age = age;

this.work = work;

this.info = function() {

alert("我叫" + this.name + "，今年" + this.age + "岁，是个" + this.work);

}

}

var Xiaosong = new Person("WooKong",23,"程序猿"); //实例化、创建对象

Xiaosong.info(); //调用info()方法

(4)用工厂方式来创建（内置对象）

var jsCreater = new Object();

jsCreater.name = "Brendan Eich"; //JavaScript的发明者

jsCreater.work = "JavaScript";

jsCreater.info = function() {

alert("我是"+this.work+"的发明者"+this.name);

}

jsCreater.info();

(5)用原型方式来创建

function Standard(){}

Standard.prototype.name = "ECMAScript";

Standard.prototype.event = function() {

alert(this.name+"是脚本语言标准规范");

}

var jiaoben = new Standard();

jiaoben.event();

(6)用混合方式来创建

function iPhone(name,event) {

this.name = name;

this.event = event;

}

iPhone.prototype.sell = function() {

alert("我是"+this.name+"，我是iPhone5s的"+this.event+"~ haha!");

}

var SE = new iPhone("iPhone SE","官方翻新机");

SE.sell();

8.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；

9.eval是做什么的？

它的功能是把对应的字符串解析成JS代码并运行；

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

（注意：在项目里面写js代码的时候，禁止使用的，因为有安全因素）

10.什么是window对象? 什么是document对象?

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

12.写一个通用的事件侦听器函数。

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。

parseInt(转换数字,转换进制2、8、10、16)

一般般使用parseInt(“010”,10) //10 parseInt(“010”,8) //8

13.事件、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;）

14.什么是闭包（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

//li节点的onclick事件都能正确的弹出当前被点击的li索引

<ul>

<li> index = 0 </li>

<li> index = 1 </li>

<li> index = 2 </li>

<li> index = 3 </li>

</ul>

<script type="text/javascript">

var nodes = document.getElementsByTagName('li');

for(i = 0;i<nodes.length;i+=1) {

nodes[i].onclick = function() {

console.log(i+1); //不使用闭包的话，值每次都是4

}(4);

}

</script>

15."use strict";是什么意思 ? 使用它的有什么好处或坏处？

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。

16.如何判断一个对象是否属于某个类？

17.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);

18.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。

19.对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"}

(注意：强调一下之前做的项目里面通过ajax请求，后端返回数据常用的格式)

20.解释一下Math常用哪些方法？

//x为传入参数

Math.abs(x); //返回数的绝对值。

Math.ceil(x); //对数进行上舍入或向上取整

Math.floor(x); //对数进行下舍入或向下取整

Math.max(x,y) //返回 x 和 y 中的最大值。

Math.min(x,y) //返回 x 和 y 中的最小值。

Math.random() // 返回 0 ~ 1 之间的随机数。

Math.round(x) //把数四舍五入为最接近的整数。

Math.PI //返回圆周率（约等于3.14159）。

21.js延迟加载的方式有哪些？

Defer(只支持 IE)和async、动态创建DOM方式（用得最多）、按需异步载入js

22.ajax是什么?

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

23.同步和异步的区别?

同步的概念应该是来自于操作系统中关于同步的概念:

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。

24.如何解决跨域问题?

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

25.模块化开发怎么做？

26.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？）

27.知道ECMAScript 6里怎么写class么?

ES6的class可以看作只是一个语法糖，它的绝大部分功能，ES5都可以做到，新的class写法只是让对象原型的写法更加清晰、更像面向对象编程的语法而已。

//定义类

class Point {

constructor(x,y) { //构造方法

this.x = x; //this关键字代表实例对象

this.y = y;

}

toString() {

return '(' + this.x + ',' + this.y + ')';

}

}

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

AMD 规范在这里：[https://github.com/amdjs/amdjs-api/wiki/AMD](https://github.com/amdjs/amdjs-api/wiki/AMD" \t "_blank)

CMD 规范在这里：[https://github.com/seajs/seajs/issues/242](https://github.com/seajs/seajs/issues/242" \t "_blank)

29.异步加载的方式有哪些？

(1)defer，只支持 IE

(2)async:

(3)创建 script，插入到 DOM 中，加载完毕后 callBack

30.call() 和 .apply() 的区别？

**一、方法的定义**   
**call方法:**   
语法：call(thisObj，Object)  
定义：调用一个对象的一个方法，以另一个对象替换当前对象。  
说明：  
call 方法可以用来代替另一个对象调用一个方法。call 方法可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。   
如果没有提供 thisObj 参数，那么 Global 对象被用作 thisObj。   
  
**apply方法：**   
语法：apply(thisObj，[argArray])  
定义：应用某一对象的一个方法，用另一个对象替换当前对象。   
说明：   
如果 argArray 不是一个有效的数组或者不是 arguments 对象，那么将导致一个 TypeError。   
如果没有提供 argArray 和 thisObj 任何一个参数，那么 Global 对象将被用作 thisObj， 并且无法被传递任何参数。

31.JavaScript中的作用域与变量声明提升？

参考地址：<http://blog.csdn.net/sunxing007/article/details/9034253>

http://www.cnblogs.com/silentjesse/p/4024536.html

32.如何编写高性能的Javascript？

参考地址：http://developer.51cto.com/art/200906/131335.htm

33.那些操作会造成内存泄漏？

内存泄漏是指任何对象在您不再拥有或需要它之后任然存在。

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）

34.JQuery的源码看过吗？能不能简单说一下它的实现原理？

35.知道什么是webkit么? 知道怎么用浏览器的各种工具来调试和debug代码么?

Chrome、safari浏览器的内核。

其中chrome的模板引擎使用的是Gecko

Gecko在互联网中是最流行的[排版引擎](http://baike.baidu.com/view/459307.htm" \t "_blank)之一，其流行程度仅次于[Trident](http://baike.baidu.com/view/682014.htm" \t "_blank)（用于Windows版的[Internet Explorer](http://baike.baidu.com/view/85144.htm" \t "_blank)4.0版本开始），其后尚有[WebKit](http://baike.baidu.com/view/1510583.htm" \t "_blank)（用于[Safari](http://baike.baidu.com/view/110484.htm" \t "_blank)以及[Google Chrome](http://baike.baidu.com/view/1835854.htm" \t "_blank)）以及[Presto](http://baike.baidu.com/view/682394.htm" \t "_blank)（用于Opera）。

可以通过js检查当前所使用浏览器的内核以及模版引擎

alert(navigator.userAgent);

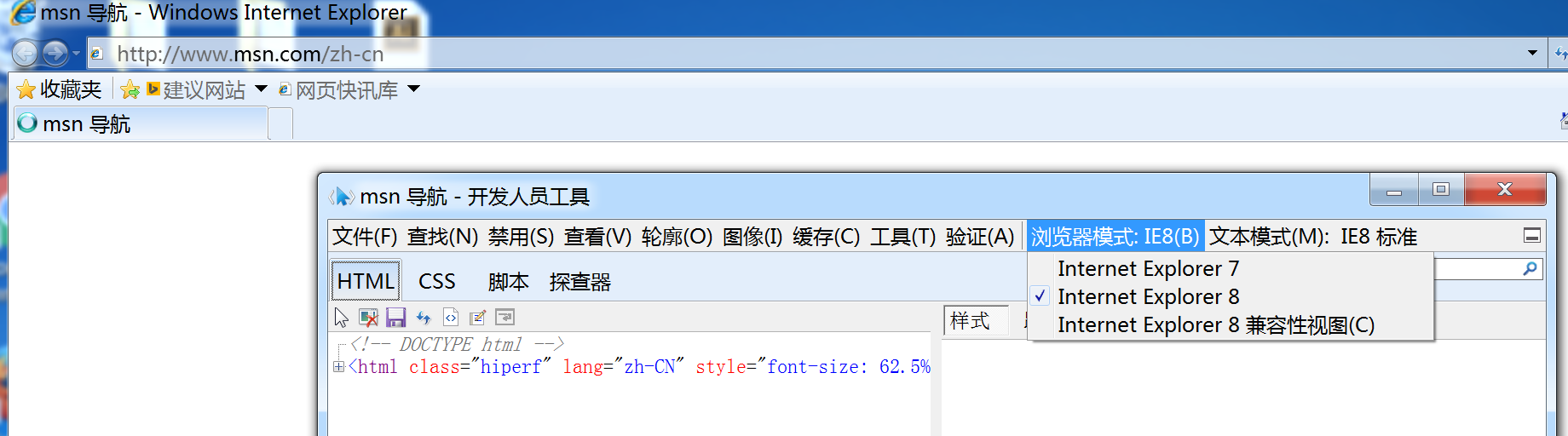
36.如何测试前端代码么? 知道BDD, TDD, Unit Test么? 知道怎么测试你的前端工程么(mocha, sinon, jasmin, qUnit..)?

如何测试前端代码么?（答案为纯属于个人经验分享如下）

PC端浏览器：市场份额：http://tongji.baidu.com/data/browser

1. 编写完前端静态html+css(注意不包括js)
2. 加上js代码以及逻辑处理（再进行测试）

在各个浏览器里面做兼容测试：1）.ietest—ie6-ie11（ie模拟浏览器）、真正的windows版本自带的IE浏览器F12可以看见下图,可以切换不同的浏览器模式，后面跟着把文档模式也改成浏览器模式一样



2).FF(火狐浏览器) 、chrome（谷歌浏览器）、opera（欧朋浏览器）、maxthon（遨游浏览器）、safari(苹果的浏览器)、猎豹安全浏览器

比较流氓的浏览器：（不知不觉电脑上就给安装上了）

360浏览器（可以切换浏览器模式）、搜狗高速浏览器、腾讯QQ浏览器、百度浏览器

还有其它用户量不大的浏览器就不用一一列举了。

http://liulanqixiazai2016.lofter.com/

移动端浏览器：移动份额

uc浏览器（知道为啥要第一个写吗？在移动端用户量最多）、腾讯qq浏览器、百度浏览器、360浏览器、猎豹浏览器、safari 、chrome、opera、maxthon、ff、搜狗浏览器、海豚浏览器

# 2016年网页浏览器发展现状分析

<http://www.chinabgao.com/k/liulanqi/22495.html>

移动设备品牌占比

http://tongji.baidu.com/data/mobile/brand

**知道BDD, TDD, Unit Test么?**

TDD：测试驱动开发（Test-Driven Development）

BDD：行为驱动开发（Behavior Driven Development）

ATDD：验收测试驱动开发（Acceptance Test Driven Development）

参考地址：<http://www.testwo.com/article/414>

**知道怎么测试你的前端工程么(mocha, sinon, jasmin, qUnit..)?**

参考地址：

<http://www.tuicool.com/articles/yuMvQz>

**sinon** 参考地址：<http://www.maobashi.com/1366.html>

参考地址：http://www.w3cmark.com/2016/446.html

37.前端templating(Mustache, underscore, handlebars)是干嘛的, 怎么用?

前端js模板引擎

Mustache：

参考地址：http://www.open-open.com/lib/view/open1416792564461.html

http://www.cnblogs.com/iyangyuan/archive/2013/12/12/3471227.html

underscore

[Underscore](http://github.com/jashkenas/underscore/) 是一个 JavaScript 工具库，它提供了一整套函数式编程的实用功能，但是没有扩展任何 JavaScript 内置对象。 他解决了这个问题：“如果我面对一个空白的 HTML 页面，并希望立即开始工作，我需要什么？” 他弥补了 [jQuery](http://jquery.com) 没有实现的功能，同时又是 [Backbone](http://backbonejs.org) 必不可少的部分。

参考地址：http://www.bootcss.com/p/underscore/

简单做个了解就可以（真实用在仔细学习）

38.简述一下 Handlebars 的基本用法？

参考地址： http://baike.baidu.com/link?url=OF4LwdRf8jbdYZt8w1Yig7Q7dSCRA2jvpfxRISOE91l6kQzl1wZwmHnJb1CEJZ4Ov7IE5wHVPvQbBam8B-O0E\_

<http://www.cnblogs.com/iyangyuan/archive/2013/12/12/3471227.html>

39.ie各版本和chrome可以并行下载多少个资源

IE6 两个并发，iE7升级之后的6个并发，之后版本也是6个

Firefox，chrome也是6个

40. 谈一谈你对 ECMAScript6 的了解？

ECMAScript 6 是JavaScript语言的下一代标准，已经在2015年6月正式发布了。它的目标，是使得JavaScript语言可以用来编写复杂的大型应用程序，成为企业级开发语言。

标准的制定者有计划，以后每年发布一次标准，使用年份作为标准的版本。因为当前版本的ES6是在2015年发布的，所以又称ECMAScript 2015。也就是说，ES6就是ES2015

## 41. 说说最近最流行的一些东西吧？常去哪些网站？

Node.js、Mongodb、npm、MVVM、MEAN、three.js,React 。

网站：w3cfuns,sf,hacknews,CSDN,慕课，博客园，InfoQ,w3cplus等

42.前端安全问题？

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

1.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。

2.避免直接在cookie 中泄露用户隐私，例如email、密码等等。 3.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。

4.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：

　　1.登录受信任网站A，并在本地生成Cookie。 　　2.在不登出A的情况下，访问危险网站B。

CSRF的防御

1. 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。
2. 使用验证码

43.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++) {}

44.Jquery与jQuery UI有啥区别？

`jQuery`是一个js库，主要提供的功能是选择器，属性修改和事件绑定等等。

`jQuery UI`则是在jQuery的基础上，利用jQuery的扩展性，设计的插件。提供了一些常用的界面元素，诸如对话框、拖动行为、改变大小行为等等

45.jQuery和Zepto的区别？各自的使用场景？

参考答案：<http://blog.csdn.net/kongjiea/article/details/42522305>

jQuery偏向于pc端

Zepto偏向于移动端

46.Zepto的点透问题如何解决？

e.preventDefault(); // 阻止“默认行为”

参考地址：

http://www.bubuko.com/infodetail-649496.html

http://my.oschina.net/chainlong/blog/225702

47.jQueryUI如何自定义组件?

48.写一个无刷新的网站，并且能在浏览器前进、后退时正确响应怎么实现？

49.如何判断当前脚本运行在浏览器还是node环境中？（阿里）

通过判断Global对象是否为window，如果不为window，当前脚本没有运行在浏览器中

50.移动端最小触控区域是多大？

51.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?

52.把 Script 标签 放在页面的最底部的body封闭之前 和封闭之后有什么区别？浏览器会如何解析它们？

参考资料：

https://segmentfault.com/a/1190000004292479?\_ea=560235

http://www.zhihu.com/question/20027966

53.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。）

参考资料：http://www.mamicode.com/info-detail-666685.html

54.Node.js的适用场景？

高并发、聊天、实时消息推送、服务器做前端资源压缩