1．7个去伪存真的JavaScript面试题

1.1创建Javascript对象的两种方法是什么？

单例模式

var o = {name:’zhangld’}

new 关键字(构造器创建对象)

function Hello(x){this.x = x}

var o = new Hello(1)//new 返回引用

追问：

new关键字做了什么？

<http://blog.liuwanlin.info/newzhi-shao-gan-liao-5jian-shi/>

function Base() {  
 this.str='aa'  
 }  
 // 1.创建空对象  
 var obj = {}  
// 2.将空对象的\_\_proto\_\_指向构造函数的prototype  
 obj.\_\_proto\_\_= Base.prototype;  
// 3.使用空对象作为上下文调用构造函数  
 Base.call(obj)

使用new关键字，什么情况下创建对象？

工厂模式

Object.create()//只能拷贝方法

var Person = {  
 name:'zhanglongde',  
 create: function (n) {  
 this.name= n  
 }  
}  
var o = Object.create(Person)  
o.create('Hi')

Object.assign()//属性和方法都能拷贝

var Person = {  
 name:'zhanglongde',  
 create: function (n) {  
 this.name= n  
 }  
}  
var o = Object.***assign***(Person)

手动工厂模式

var Person = (function (p) {  
 p.name = 'zhanglongde'  
  
 p.init = function (n) {  
 p.name = n  
 }  
 return p  
})(Person || {})  
  
window.onload = function () {  
 Person.init('zld')  
}

Class Point{

Constructor(x,y){this.x= x;this.y=y} //通过new命令生成对象实例时，自动调用该方法

}

1.2如何创建数组？

JS数组是稀疏的

使用方括号创建

Var arr = []

Var a = Array()

数组去重方法<http://web.jobbole.com/83425/>

Array.from(new Set(arr))

[…new Set(arr)]

叹为观止的ES6 Array

1.3什么是变量提升

变量声明提升，赋值不提升

Var 和 function提升

提升到作用域头部

1.4全局变量有什么风险，以及如何保护代码不受干扰？

变量名污染

工厂模式

Var o = (funcstion(o){ …. return o;})(o || {})

1.5如何通过Javascript对象中的成员变量迭代？

For (key in obj){

if (obj.hasOwnProperty(key)) {

Obj[key]

}

}

1.6什么是闭包？

有权访问另一函数作用域内变量的函数

1.7请描述你经历过的JAV阿Script单元测试

2. 80%应聘者都不及格的JS面试题

for(var i = 0; i<5; i++){  
 setTimeout(function () {  
 console.log(new Date, i)  
 },1000)  
}  
console.log(new Date, i)

3. 我是如何同时拿到阿里和腾讯offer的

腾讯面试题

1. 请介绍你的项目
2. XMLHttpRequest

创建XHR

连接服务器

发送

回调

1. Http状态码

200 204

304 Not modified

400参数错误 404

500服务器内部错误

1. Cache-control
2. CSS动画原理

补间动画 缓动曲线

逐帧动画 step

Js 物理动画 摩擦力

1. 圣杯布局

三栏布局 浮动

如何使 中间布局在最前面显示 用到负边距

1. ie的某些兼容性

浮动ie产生的双倍距离

1. 项目
2. 跨域

Hack方法：JSONP window.name 图像ping

非hack方法 CORS postMessage SSE websocket sessionStorage

1. JavaScript数据类型
2. 页面加载

输入地址到呈现

Window.onload $()

1. 字符串转化
2. Jsonp原理

JSON外层过上函数

1. CSS合并方法
2. 盒子模型
3. 定位

淘宝面试题

1. 自我介绍
2. 事件委托
3. 模块化工具的特点
4. Session
5. Ajax跨域
6. This
7. 长期规划
8. 前端怎么实现模块化
9. 最熟悉的框架
10. 前后端协同开发

前后端分离 AJAX websocket

服务端渲染

1. 前端学习路线
2. 基础算法
3. Javascript动画算法
4. 拖拽实现

原生js mouseenter mouseover mouseend

H5 dragstart dragend

Dragenter进入目标元素 dragleave离开目标元素

Dragover拖拽元素在目标元素上移动 必须阻止默认行为

Drop拖拽元素在目标元素上，同时松开鼠标

1. Javascript继承实现

前端性能优化

1. 文件合并
2. CSS精灵
3. 避免页面跳转
4. Javascript延迟加载
5. 减少DOM元素数量：会加大页面加载和脚本执行的效率
6. 浏览器一般对同一个域的下载链接数有所限制，按照域名划分下载内容可以使浏览器增大并行下载链接，但是主语控制域名使用在2-4个之间，不然dns查询也是个问题
7. 一般网站规划会将静态资源放在类似于static.example.com,动态内容放在[www.example.com](http://www.example.com)上。这样做还有一个好处是可以在静态的域名商避免使用cookie.后面我们会在cookie的规则中提到。

每次客户端请求都会带上cookie

1. 404我们都不陌生，代表服务器没有找到资源，我们要特别注意404的情况不要在我们提供的网页资源上，客户端发送一个 请求但是服务器却返回一个无用的结果，时间浪费掉了。
2. 使用CDN
3. 使用缓存:cache-control和expires
4. Gzip压缩传输文件
5. 配置etag
6. 避免空的src：空的图片src仍然会使浏览器发送请求到服务器，这样完全是浪费时间，而且浪费服务器的资源。尤其是你的网站每天被很多人访问的时候，这种空请求造成的上海不容忽略。
7. 减小cookie大小：cookie被用来做认证或个性化设置，其信息被包含在http报文头中，对于cookie我们要注意以下几点，来提高请求的响应速度
8. CSS样式表重置：将样式表(css)放在网页和HEAD中会让网页显示加载速度更快，因为这样做可以使浏览器逐步加载已将下载的网页内容。这对内容比较多的网页尤其重要，用户不用一直等待在一个白屏上，而是可以先看已经下载的内容。

如果将样式表放在底部，浏览器会拒绝渲染已经下载的网页，因为大多数浏览器在实现时都努力避免重绘，样式表中的内容是绘制网页的关键信息，没有下载下来之前只好对不起观众了。

1. CMD vs AMD

CommonJS是服务器端模块的规范，node.js采用了这个规范。

根据CommonJS规范，一个单独的文件就是一个模块，也就是说，在该模块内部定义的变量，无法被其他模块读取，除非定义为global对象的属性。

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。

CMD:sea.js

ADM规范则是非同步加载模块，允许制定回调函数。由于Node.js主要用于服务器编程，模块文件一般都已经存在于本地磁盘，所以加载起来比较快，不用考虑非同步加载的方式，不用考虑非同步加载的方式，所以CommonJS规范比较适用。

但是，如果是浏览器环境，要从服务器端加载模块，这是就必须采用非同步模式，因此浏览器端一般采用AMD规范。

AMD:require.js 异步

前端安全

CSRF

简介：跨站请求为战。

攻击者盗用了你的身份，以你的名义发送恶意请求。他可以做到的事情包括：以你的名义发送邮件、发消息、盗取你的账号、甚至购买商品…造成的问题包括：个人隐私泄露以及财产安全。

原理：登录受信任的网站A，并且在本地生成cookie，在不登出A网站的情况下，访问危险网站B。

常见类型：

1. 使用get更新请求资源，容易在B使用img标签造成攻击
2. 前端虽然改成了post请求，但是B网站通过iframe等手段同样造成了攻击

防御机制

1. 验证码：用户每次提交都需要在表单中填写图片上一个随机的字符串
2. 添加隐藏域token:添加一个隐藏的输入框，包含token，服务端验证是否匹配添加
3. 使用HTTP refer头部进行判断，如果不是业务域名发起的HTTP请求，直接过滤。

XSS

XSS是什么？他的全名是：cross-site scripting,为了和CSS层叠样式区分所以取名XSS。是一种网站应用程序的安全漏洞攻击，是代码注入的一种。

它允许恶意用户将代码注入到网页上，其他用户在观看网页时就会受到影响，这类攻击通常包含了HTML以及用户端脚本语言。

例子：