**前端面试题**

[常见知识点 2](#_Toc23606)

[1. 浏览器的内核分为几种？有什么分别？ 3](#_Toc7222)

[2. 前端页面有哪三层构成，分别是什么？作用是什么？ 3](#_Toc9331)

[3. 对于网页布局掌握怎么样?标签语意了解吗？ 3](#_Toc10996)

[4. 你如何对网站的文件和资源进行优化？期待的解决方案包括： 3](#_Toc4599)

[5. 什么是语义化的HTML？ 3](#_Toc9417)

[6. 清除浮动的几种方式，各自的优缺点 3](#_Toc11892)

[7. 前端优化知识都知道哪些？ 4](#_Toc15875)

[8. HTTP协议的状态消息都有哪些?(如200、302对应的描述)。 4](#_Toc28885)

[9. 都使用和了解过哪些编辑器?都使用和了解过哪些日常工具? 4](#_Toc15951)

[10. <!DOCTYPE>标签的定义与用法。 5](#_Toc17599)

[HTML部分 5](#_Toc17091)

[1. html空元素有哪些？ 5](#_Toc15083)

[2. 对WEB标准以及W3C的理解与认识 5](#_Toc18759)

[3. xhtml和html有什么区别 6](#_Toc4460)

[4. HTML5都有哪些新的JS API？ 6](#_Toc6208)

[Css部分 6](#_Toc31528)

[1. html5 的CSS3新增的伪类？ 6](#_Toc27825)

[2. CSS3都有哪些新特性呢？ 7](#_Toc2056)

[3. 常见浏览器兼容性问题与解决方案 7](#_Toc27170)

[4. 行内元素有哪些？块级元素有哪些？CSS的盒模型？（一般笔试题可能考到） 9](#_Toc23461)

[5. CSS引入的方式有哪些? link和@import的区别是? 9](#_Toc25365)

[6. CSS选择符有哪些？哪些属性可以继承？优先级算法如何计算？内联和important哪个优先级高？ 9](#_Toc3722)

[7、css的基本语句构成是？ 9](#_Toc1489)

[8、双边距BUG float引起的 9](#_Toc13878)

[9、像素问题 使用float引起的 9](#_Toc6342)

[10、超链接hover 点击后失效 10](#_Toc12936)

[11、Ie z-index问题 10](#_Toc2781)

[12、Png 透明 10](#_Toc4928)

[13、Min-height 最小高度 ！ 10](#_Toc32616)

[14、select 在ie6下遮盖 10](#_Toc16043)

[15、为什么没有办法定义1px左右的宽度容器 10](#_Toc31388)

[16.<img>标签上title与alt属性的区别是什么？ 10](#_Toc2762)

[12.描述css reset的作用和用途。 10](#_Toc15024)

[13.解释css sprites，如何使用。 10](#_Toc17505)

[14.浏览器标准模式和怪异模式之间的区别是什么？ 11](#_Toc27413)

[15、介绍所知道的CSS hack技巧(如：\_， \*， +， \9， !important 之类)。 11](#_Toc15318)

[16、CSS层叠是什么？介绍一下。 12](#_Toc2667)

[JavaScript部分 12](#_Toc28113)

[1. Document.write 和innerTHML区别 12](#_Toc24860)

[2. 对JSON的理解： 13](#_Toc19953)

[JSON 语法规则 13](#_Toc5570)

[3. Null和undefined区别： 14](#_Toc11284)

[4. javascript的typeof返回哪些数据类型 14](#_Toc5567)

[5. 例举3种强制类型转换和2种隐式类型转换？ 14](#_Toc12553)

[6. split() join() 的区别 14](#_Toc31676)

[7. 数组方法pop() push() unshift() shift() 14](#_Toc5538)

[8. 事件绑定和普通事件有什么区别 14](#_Toc10977)

[9. IE和DOM事件流的区别 15](#_Toc3967)

[10. IE和标准下有哪些兼容性的写法 15](#_Toc19716)

[11. ajax请求的时候get 和post方式的区别 16](#_Toc2314)

[12. call和apply的区别 16](#_Toc4901)

[13. ajax请求时，如何解释json数据 16](#_Toc9853)

[14. 事件委托是什么 16](#_Toc21229)

[15. 如何阻止事件冒泡和默认事件 16](#_Toc26599)

[16. DOM操作 - 怎样添加、移除、移动、复制、创建和查找节点(这个问题真心是基础题，一般不会问)。 16](#_Toc12558)

[17. 解释jsonp的原理，以及为什么不是真正的ajax 17](#_Toc30858)

[18. javascript的本地对象，内置对象和宿主对象 17](#_Toc18355)

[19. document load 和document ready的区别 17](#_Toc32271)

[20. ”==”和“===”的不同 17](#_Toc29028)

[21. javascript的同源策略 18](#_Toc3675)

[22. 简述JavaScript封装。 18](#_Toc25604)

[23. 瀑布流布局或者流式布局是否有了解 18](#_Toc26893)

[24. JavaScript继承有哪两种形式形式，进行描述。 18](#_Toc25817)

[25. 在JS中this关键字的使用场合和用法(如在构造函数中、setTimeout中等)。 19](#_Toc14828)

[jQuery相关 19](#_Toc26046)

[1.jQuery源码是否尝试去读过？说说基本的架构或者 jQuery.fn.init 中都做了哪些判断。 19](#_Toc5074)

[2. 都知道哪些不好的jQuery书写方式。 20](#_Toc7404)

[超出范围 20](#_Toc4048)

[1． 同步异步区别 20](#_Toc29691)

[2． JavaScript继承有哪两种形式，进行描述。 20](#_Toc6823)

[3． 简述下cookie的操作，还有cookie的属性都知道哪些。 21](#_Toc20494)

[4． IE与FF的JS兼容性都知道哪些。 22](#_Toc2758)

[5． jquery 闭包 是什么意思 有什么用呢 22](#_Toc1099)

[6． 闭包是什么意思？ 22](#_Toc11110)

[7． AJAX是什么? AJAX的交互模型(流程)? AJAX跨域的解决办法? 23](#_Toc2307)

[8． 正则表达式有系统学习过吗(看书或网上教程)？有的话就问问简单点的邮箱验证、URL验证， 或者问问 贪婪匹配与懒惰匹配 的理论知识。 24](#_Toc3468)

# 常见知识点

## 浏览器的内核分为几种？有什么分别？

Trident内核：IE最先开发或使用的，也称IE内核，360浏览器使用的也是IE内核；

Webkit内核：谷歌chrome浏览器最先开发或使用，也叫谷歌内核，枫树浏览器、太阳花使用的也是谷歌内核；

Gecko内核： Netscape6开始采用的内核，后来的Mozilla FireFox (火狐浏览器) 也采用了该内核，K-Meleon浏览器也是使用这种内核；

Presto内核：目前只有Opera浏览器采用该内核  
  
此外，由于IE浏览器在国内的普及率非常高，所以造成了很多网上银行和支付系统只支持IE的Trident内核，其他浏览器访问根本无法进行正常支付和转账等业务。这就造成了经常在网上购物的人离不开Trident内核的IE浏览器。但是Trident内核的速度和兼容性上又是在比不上Chrome浏览器的WebKit内核和Firefox浏览器的Gecko内核。于是双核浏览器应运而生，比如傲游、枫树浏览器（ChromePlus）、搜狗高速浏览器。 求采纳！

## 前端页面有哪三层构成，分别是什么？作用是什么？

结构层 Html 表示层 CSS 行为层 js

## 对于网页布局掌握怎么样?标签语意了解吗？

语义化的主要目的就是让大家直观的认识标签(markup)和属性(attribute)的用途和作用。语义化的网页的好处，最主要的就是对搜索引擎友好，有了良好的结构和语义你的网页内容自然容易被搜索引擎抓取，你网站的推广便可以省下不少的功夫。

## 你如何对网站的文件和资源进行优化？期待的解决方案包括：

文件合并

文件最小化/文件压缩

使用CDN托管

缓存的使用

## 什么是语义化的HTML？

直观的认识标签 对于搜索引擎的抓取有好处

## 清除浮动的几种方式，各自的优缺点

1.使用空标签清除浮动 clear:both（理论上能清楚任何标签，，，增加无意义的标签）

2.使用overflow:auto（空标签元素清除浮动而不得不增加无意代码的弊端,,使用zoom:1用于兼容IE）

3.是用afert伪元素清除浮动(用于非IE浏览器)

## 前端优化知识都知道哪些？

图片优化，  
CSS优化  
JS优化等等

一、提倡前端开发工程师在书写xhtml的时候做到结构语义化。

二、减少请求文件的大小，压缩css、js文件，在线javascript compress。

三、背景图片数量及大小的优化， 采用sprite拼接图片。

四、内容图片的大小的优化

详情登录：<http://www.cnblogs.com/mofish/archive/2010/10/12/1849041.html> 查看

a  减少http请求，合并css、js文件，apache的ant。

d  如果有广告栏之类的模块，用iframe。

e  将js文件放到末尾，这个页面显示就不必等js文件加载完以后再显示，也就是页面不会出现空白状态。

## HTTP协议的状态消息都有哪些?(如200、302对应的描述)。

答：Web服务器用来告诉客户端，发生了什么事，也就是一种快速解决问题的方式。

200：OK，服务器成功处理了请求。

301/302：Moved Permanently(重定向)，请求的URL已移走，一个是永久转移，一个是临时转移。Response中应该包含一个Location URL, 说明资源现在所处的位置。

304：Not Modified（未修改），客户的缓存资源是最新的，要客户端使用缓存。

404：HTTP/1.1 404 Not Found，服务器无法找到所请求的URL。

501：Internal Server Error，服务器遇到一个错误，使其无法对请求提供服务。

503：Service Unavailable（未提供此服务），服务器目前无法为请求服务服务，但过一段时间就可以恢复服务。

## 都使用和了解过哪些编辑器?都使用和了解过哪些日常工具?

答：我一般采用webStorm（网络风暴），是[jetbrains](http://baike.baidu.com/view/1980886.htm)公司开发的。

我选择它的主要原因是它能自动缩进和自动闭合，但是添加url和href的方式我不是很喜欢。要自己写文件名，但是也还能接受，因为只要你输对一个单词就能自定弹出对应文件名。

DW添加地址方式很好，但是不能自动缩进和自动闭合，我不喜欢。

## <!DOCTYPE>标签的定义与用法。

答：定义：

<!DOCTYPE>声明位于文档中的最前面的位置，处于<html>标签之前。此标签可告知浏览器文档使用哪种HTML或XHTML规范**。**

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版本。**（**假如文档中的标记不遵循doctype声明所指定的DTD，这个文档除了不能通过代码校验之外，还有可能无法在浏览器中正确显示。）

用法：

a.如果需要干净的标记，免于表现层的混乱，用XHTML Strict DTD类型。

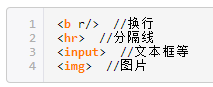
b.Transitional DTD 可包含 W3C 所期望移入样式表的呈现属性和元素. 如果用户使用了不支持层叠样式表（CSS）的浏览器以至于你不得不使用 HTML 的呈现特性时，用Transitional DTD 类型。

c.Frameset DTD 被用于带有框架的文档。除 frameset 元素取代了 body 元素之外，Frameset DTD 等同于 Transitional DTD。

# HTML部分

# **html空元素有哪些？**

没有内容的 HTML 内容被称为空元素。空元素是在开始标签中关闭的。



# **对WEB标准以及W3C的理解与认识**

标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外链css和js脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所

访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性；

# **xhtml和html有什么区别**

HTML是一种基本的WEB网页设计语言，XHTML是一个基于XML的置标语言

最主要的不同：

XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

标签名必须用小写字母。

XHTML 文档必须拥有根元素。

# **HTML5都有哪些新的JS API？**

多媒体：video、audio、  
游戏：canvas、webgl、  
存储：localstorage、sessonstorage、websql、indexedDB  
网络：websocket  
还有一些，但由于HTML5还没有定稿，其他大多是几个浏览器引擎自己加了私有扩展的实现。

答：document.querySelector("selector");

selector:根据CSS选择器返回第一个匹配到的元素，如果没有匹配到，则返回null。

document.querySelectorAll("selector");  
selector:根据CSS选择器返回所有匹配到的元素数组，如果没有匹配到，则返回空数组;

document.getElementsByClassName("selector");  
selector:根据类选择器返回所有匹配到的元素数组，如果没有匹配到，则返回空数组。

# Css部分

## html5 的CSS3新增的伪类？

1. **伪类选择器:nth-child(n)**
2. **：before 和 ：after**

## CSS3都有哪些新特性呢？

1. 第 1 选择器  
   第 2 RGBA和透明度  
   第 3 多栏布局  
   第 4 多背景图  
   第 5 Word Wrap  
   第 6 文字阴影  
   第 7 @font-face属性  
   第 8 圆角(边框半径)  
   第 9 边框图片  
   第 10 盒阴影  
   第 11 盒子大小  
   第 12 媒体查询  
   第 13 语音

css3：box-shadow阴影，border-radius圆角，transform中skew扭曲度，rotate旋转，scale缩放等，transition中all 0.5s ease 0s过渡时间。before和after。

## [常见浏览器兼容性问题与解决方案](http://blog.csdn.net/chuyuqing/article/details/37561313)

浏览器兼容问题一：不同浏览器的标签默认的外补丁和内补丁不同

问题症状：随便写几个标签，不加样式控制的情况下，各自的margin 和padding差异较大。

解决方案：CSS里    \*{margin:0;padding:0;}

备注：这个是最常见的也是最易解决的一个浏览器兼容性问题，几乎所有的CSS文件开头都会用通配符\*来设置各个标签的内外补丁是0。

浏览器兼容问题二：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

问题症状:常见症状是IE6中后面的一块被顶到下一行

碰到频率：90%（稍微复杂点的页面都会碰到，float布局最常见的浏览器兼容问题）

解决方案：在float的标签样式控制中加入 display:inline;将其转化为行内属性

备注：我们最常用的就是div+CSS布局了，而div就是一个典型的块属性标签，横向布局的时候我们通常都是用div float实现的，横向的间距设置如果用margin实现，这就是一个必然会碰到的兼容性问题。

浏览器兼容问题三：设置较小高度标签（一般小于10px），在IE6，IE7，遨游中高度超出自己设置高度

问题症状：IE6、7和遨游里这个标签的高度不受控制，超出自己设置的高度

解决方案：给超出高度的标签设置overflow:hidden;或者设置行高line-height 小于你设置的高度。

备注：这种情况一般出现在我们设置小圆角背景的标签里。出现这个问题的原因是IE8之前的浏览器都会给标签一个最小默认的行高的高度。即使你的标签是空的，这个标签的高度还是会达到默认的行高。

浏览器兼容问题四：行内属性标签，设置display:block后采用float布局，又有横行的margin的情况，IE6间距bug

问题症状：IE6里的间距比超过设置的间距

解决方案：在display:block;后面加入display:inline;display:table;

备注：行内属性标签，为了设置宽高，我们需要设置display:block;(除了input标签比较特殊)。在用float布局并有横向的margin后，在IE6下，他就具有了块属性float后的横向margin的bug。不过因为它本身就是行内属性标签，所以我们再加上display:inline的话，它的高宽就不可设了。这时候我们还需要在display:inline后面加入display:talbe。

浏览器兼容问题五：图片默认有间距

问题症状：几个img标签放在一起的时候，有些浏览器会有默认的间距，加了问题一中提到的通配符也不起作用。

解决方案：使用float属性为img布局

备注：因为img标签是行内属性标签，所以只要不超出容器宽度，img标签都会排在一行里，但是部分浏览器的img标签之间会有个间距。去掉这个间距使用float是正道。（我的一个学生使用负margin，虽然能解决，但负margin本身就是容易引起浏览器兼容问题的用法，所以我禁止他们使用）

浏览器兼容问题六：标签最低高度设置min-height不兼容

问题症状：因为min-height本身就是一个不兼容的CSS属性，所以设置min-height时不能很好的被各个浏览器兼容

解决方案：如果我们要设置一个标签的最小高度200px，需要进行的设置为：{min-height:200px; height:auto !important; height:200px; overflow:visible;}

备注：在B/S系统前端开时，有很多情况下我们又这种需求。当内容小于一个值（如300px）时。容器的高度为300px；当内容高度大于这个值时，容器高度被撑高，而不是出现滚动条。这时候我们就会面临这个兼容性问题。

浏览器兼容问题七：透明度的兼容CSS设置

做兼容页面的方法是：每写一小段代码（布局中的一行或者一块）我们都要在不同的浏览器中看是否兼容，当然熟练到一定的程度就没这么麻烦了。建议经常会碰到兼容性问题的新手使用。很多兼容性问题都是因为浏览器对标签的默认属性解析不同造成的，只要我们稍加设置都能轻松地解决这些兼容问题。如果我们熟悉标签的默认属性的话，就能很好的理解为什么会出现兼容问题以及怎么去解决这些兼容问题。

/\* CSS hack\*/

我很少使用hacker的，可能是个人习惯吧，我不喜欢写的代码IE不兼容，然后用hack来解决。不过hacker还是非常好用的。使用hacker我可以把浏览器分为3类：IE6 ；IE7和遨游；其他（IE8 chrome ff safari opera等）

◆IE6认识的hacker 是下划线\_ 和星号 \*

◆IE7 遨游认识的hacker是星号 \*

比如这样一个CSS设置：

height:300px;\*height:200px;\_height:100px;

IE6浏览器在读到height:300px的时候会认为高时300px；继续往下读，他也认识\*heihgt， 所以当IE6读到\*height:200px的时候会覆盖掉前一条的相冲突设置，认为高度是200px。继续往下读，IE6还认识\_height,所以他又会覆盖掉200px高的设置，把高度设置为100px；

IE7和遨游也是一样的从高度300px的设置往下读。当它们读到\*height200px的时候就停下了，因为它们不认识\_height。所以它们会把高度解析为200px，剩下的浏览器只认识第一个height:300px;所以他们会把高度解析为300px。因为优先级相同且想冲突的属性设置后一个会覆盖掉前一个，所以书写的次序是很重要的。

## 行内元素有哪些？块级元素有哪些？CSS的盒模型？（一般笔试题可能考到）

块级元素：div p h1 h2 h3 h4 form ul

行内元素: a b br i span input select

Css盒模型:内容，border ,margin，padding

## CSS引入的方式有哪些? link和@import的区别是?

内联 内嵌 外链 导入

区别 ：同时加载

前者无兼容性，后者CSS2.1以下浏览器不支持

Link 支持使用javascript改变样式，后者不可

## CSS选择符有哪些？哪些属性可以继承？优先级算法如何计算？内联和important哪个优先级高？

标签选择符 类选择符 id选择符

继承不如指定 Id>class>标签选择

important优先级高

## 7、css的基本语句构成是？

选择器{属性1:值1;属性2:值2;……}

## 8、双边距BUG float引起的

使用display

## 9、像素问题 使用float引起的

使用dislpay:inline -3px

## 10、超链接hover 点击后失效

使用正确的书写顺序 link visited hover active

## 11、Ie z-index问题

给父级添加position:relative

## 12、Png 透明

使用js代码 改

## 13、Min-height 最小高度 ！

Important 解决’

## 14、select 在ie6下遮盖

使用iframe嵌套

## 15、为什么没有办法定义1px左右的宽度容器

（IE6默认的行高造成的，使用over:hidden,zoom:0.08 line-height:1px）

## 16.<img>标签上title与alt属性的区别是什么？

Alt 当图片不显示是 用文字代表。

Title 为该属性提供信息

## 12.描述css reset的作用和用途。

Reset重置浏览器的css默认属性 浏览器的品种不同，样式不同，然后重置，让他们统一

## 13.解释css sprites，如何使用。

Css 精灵 把一堆小的图片整合到一张大的图片上，减轻服务器对图片的请求数量

## 14.浏览器标准模式和怪异模式之间的区别是什么？

盒子模型 渲染模式的不同

使用 window.top.document.compatMode 可显示为什么模式

## 15、介绍所知道的CSS hack技巧(如：\_， \*， +， \9， !important 之类)。

由于不同的浏览器对CSS的支持及解析结果不一样，还由于CSS中的优先级的关系。我们就可以根据这个来针对不同的浏览器来写不同的CSS。

比如 IE6能识别下划线“\_”和星号“ \* ”，IE7能识别星号“ \* ”，但不能识别下划线“\_”，而firefox两个都不能认识。等等

**CSS hack：**不同的浏览器，对CSS的解析认识不一样，因此会导致生成的页面效果不一样，得不到我们所需要的页面效果。这个时候我们就需要针对不同的浏览器去写不同的CSS，**让它能够同时兼容不同的浏览器，**能在不同的浏览器中也能得到我们想要的页面效果。这个针对不同的浏览器写不同的CSS code的过程，就叫CSS hack，也叫写**CSS hack。**

以下是一些常用的CSS兼容技巧

1）火狐下给div设置padding后会导致width和height 增加， 但IE不会。（可用！important解决）

2）垂直居中，将 line-height设置为当前div相同的高度， 再通过vertical-align： middle；（ 注意内容不要换行）

3）水平居中，margin:0 auto;（当然不是万能）

4）若需给a标签内内容加上样式， 需要设置 display： block;（常见于导航标签）

5）浮动IE产生的双倍距离

在IE下，当一个div设置了float后，然后给他设置margin，就会出现加倍的margin，解决的办法是给div设置display:inline。

## 16、CSS层叠是什么？介绍一下。

层叠指的是样式的优先级，当产生冲突时以优先级高的为准。  
1. 开发者样式>读者样式>浏览器样式（除非使用!important标记 ）  
2. id选择符>（伪）类选择符>元素选择符  
3. 权重相同时取后面定义的样式

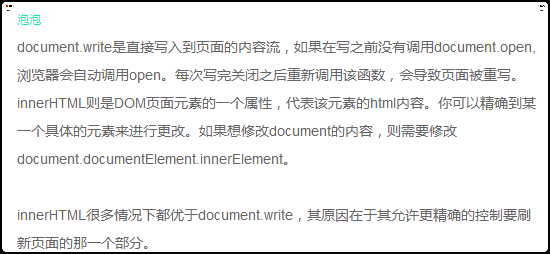
有时会被问到些刁钻点的题，比如position值都有哪些，CSS3都有哪些新内容...

# JavaScript部分

## Document.write 和innerTHML区别

document.write()实在页面中出入内容；如果不指定元素的话；它会覆盖掉整个页面内容

innerHTML只是修稿Dom元素的内容包含该元素下子元素结构



## 对JSON的理解：

json简单说就是javascript中的对象和数组，所以这两种结构就是对象和数组两种结构，通过这两种结构可以表示各种复杂的结构

**对象：**对象在js中表示为“{}”括起来的内容，数据结构为 {key：value,key：value,...}的键值对的结构，在面向对象的语言中，key为对象的属性，value为对应的属性值，所以很容易理解，取值方法为 对象.key 获取属性值，这个属性值的类型可以是 数字、字符串、数组、对象几种。

**2、数组：**数组在js中是中括号“[]”括起来的内容，数据结构为 ["java","javascript","vb",...]，取值方式和所有语言中一样，使用索引获取，字段值的类型可以是 数字、字符串、数组、对象几种。

经过对象、数组2种结构就可以组合成复杂的数据结构了。

**格式**

## **JSON 语法规则**

JSON 语法是 JavaScript 对象表示法语法的子集。

* 数据在名称/值对中
* 数据由逗号分隔
* 花括号保存对象
* 方括号保存数组

例子：

{"firstName":"Brett"}

{"firstName":"Brett","lastName":"McLaughlin","email":"aaaa"}

## Null和undefined区别：

Undefined（没有定义，不知道是什么东西 ） 表示变量声明但未初始化时的值，

null（一个盒子，是空的，里面没有放值。 ）表示准备用来保存对象，还没有真正保存对象的值。从逻辑角度看，null值表示一个空对象指针。

## javascript的typeof返回哪些数据类型

Object number function boolean undefine

## 例举3种强制类型转换和2种隐式类型转换？

强制（parseInt,parseFloat,number）

隐式（== - ===）

## split() join() 的区别

前者是切割成数组的形式，后者是将数组转换成字符串

## 数组方法pop() push() unshift() shift()

Push()尾部添加 pop()尾部删除

Unshift()头部添加 shift()头部删除

## 事件绑定和普通事件有什么区别

普通添加事件的方法不支持添加多个事件，最下面的事件会覆盖上面的，而事件绑定（addEventListener）方式添加事件可以添加多个。

举例：

## IE和DOM事件流的区别

1.执行顺序不一样、

2.参数不一样

3.事件加不加on

4.this指向问题

## IE和标准下有哪些兼容性的写法

Var ev = ev || window.event

document.documentElement.clientWidth || document.body.clientWidth

Var target = ev.srcElement||ev.target

## ajax请求的时候get 和post方式的区别

一个在url后面 一个放在虚拟载体里面

有大小限制

安全问题

应用不同 一个是论坛等只需要请求的，一个是类似修改密码的

## call和apply的区别

Object.call(this,obj1,obj2,obj3)

Object.apply(this,arguments)

## ajax请求时，如何解释json数据

使用eval parse 鉴于安全性考虑 使用parse更靠谱

## 事件委托是什么

让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行！

## 如何阻止事件冒泡和默认事件

canceBubble return false

## DOM操作 - 怎样添加、移除、移动、复制、创建和查找节点(这个问题真心是基础题，一般不会问)。

答：添加：append,prepend,after，before等等。

删除：remove，empty。

替换：replaceWith，replaceAll。

复制：clone。

创建：creatElement,createTextNode。

查找：getElmentById,getElmentsByTagName(通过标签name值),getElmentByName(通过元素name值)。

obj.appendChidl()

obj.innersetBefore

obj.replaceChild

obj.removeChild

## 解释jsonp的原理，以及为什么不是真正的ajax

动态创建script标签，回调函数

Ajax是页面无刷新请求数据操作

## javascript的本地对象，内置对象和宿主对象

本地对象为array obj regexp等可以new实例化

内置对象为gload Math 等不可以实例化的

宿主为浏览器自带的document,window 等

## document load 和document ready的区别

Document.onload 是在结构和样式加载完才执行js

Document.ready原生种没有这个方法，jquery中有 $().ready(function)

## ”==”和“===”的不同

”==”与”===”是不同的,一个是判断值是否相等,一个是判断值及类型是否完全相等。

前者会自动转换类型（如果一个值是数字，另一个值是字符串，把字符串转换为数字，再用转换后的值进行比较。）

后者不会

## javascript的同源策略

一段脚本只能读取来自于同一来源的窗口和文档的属性，这里的同一来源指的是主机名、协议和端口号的组合

## 简述JavaScript封装。

　　其它相关的加分项：

都使用和了解过哪些编辑器？都使用和了解过哪些日常工具？

都知道有哪些浏览器内核？开发过的项目都兼容哪些浏览器？

国内外的JS牛人都知道哪些？

## 瀑布流布局或者流式布局是否有了解

瀑布流，又称瀑布流式布局。是比较流行的一种网站页面布局，视觉表现为参差不齐的多栏布局，随着页面滚动条向下滚动，这种布局还会不断加载数据块并附加至当前尾部。最早采用此布局的网站是Pinterest，逐渐在国内流行开来。国内大多数清新站基本为这类风格，像花瓣网、蘑菇街、美丽说等。

瀑布流布局：采用绝对定位来给每张图片或者模块定位。

流式布局：采用浮动式给模块定位。可以做出响应式布局。

## JavaScript继承有哪两种形式形式，进行描述。

构造函数借用：在子类构造方法内，通过apply/call将this作为参数传入。

优点：

　　可以向父类构造方法传递参数，即给apply第二个参数：arguments；

　　父类中的属性都被复制到子类实例中，属性之间无干扰，无论是引用类型还是封装类型。

缺点：

　　每一个实例对象都拥有一份父类方法的拷贝，互不干扰，所以无法统一修改；

　　无法拥有父类原型中定义的方法；

　　子类的实例对象不能通过 instanceof 操作符判断是否是父类的实例。

原型链：指定子类的prototype为父类的一个实例对象。

优缺点和构造函数借用刚好相反。这里特别说明下属性之间相互干扰（对应构造函数借用的优点2）。

组合式继承：上面两种方式互补一下，即用构造方法借用来继承父类属性，用原型链来继承父类方法。

优点：

　　封装保护了内部数据的完整性；

　　封装使对象的重构更轻松；

　　弱化模块间的耦合，提高对象的可重用性；

　　有助于避免命名空间冲突。

缺点：

　　私用方法很难测试；

　　必须与复杂的作用域链打交道，使错误调度更困难；

　　容易形成过度封装；

　　JavaScript并不原生支持封装，所以在JavaScript中实现封装存在复杂性的问题。

## 在JS中this关键字的使用场合和用法(如在构造函数中、setTimeout中等)。

# jQuery相关

## 1.jQuery源码是否尝试去读过？说说基本的架构或者 jQuery.fn.init 中都做了哪些判断。

jQuery.fn.init 中都做了哪些判断：http://www.jb51.net/article/50813.htm

## 都知道哪些不好的jQuery书写方式。

答：定义变量时，全局变量不添加var，因为这个是删除不掉的。还有就是像写流水账一样的编程方式，将很多东西都蹂躏在一起，没有将不同的对象和功能给区分开。定义一个对象最好是以函数表达式的方式定义，尽量不要定义对象字面量，这样里面的私有属性或者方法容易被用户修改，也就是不安全。

# 超出范围

## **同步异步区别**

javascript 异步表示async，指：代码执行不按顺序，‘跳过’执行，待其他某些代码执行完后，再来执行，称为“异步”。

javascript同步表示sync，指：代码依次执行。

就是这么回事

javascript所谓的“线程”，就是这样一种概念

这种情况容易出现在 ajax 当中，因为ajax最可能，也最多用到async 或者 sync的概念属性。

**同步：**脚本会停留并等待服务器发送回复然后再继续。提交请求->等待服务器处理->处理完毕返回，这个期间客户端浏览器不能干任何事。

**异步：**脚本允许页面继续其进程并处理可能的回复。请求通过事件触发->服务器处理（这是浏览器仍然可以作其他事情）->处理完毕

若要在使用ajax请求后处理发送请求返回的结果，最好使用同步请求。

## **JavaScript继承有哪两种形式，进行描述。**

答：构造函数借用：在子类构造方法内，通过apply/call将this作为参数传入。

优点：

　　可以向父类构造方法传递参数，即给apply第二个参数：arguments；

　　父类中的属性都被复制到子类实例中，属性之间无干扰，无论是引用类型还是封装类型。

缺点：

　　每一个实例对象都拥有一份父类方法的拷贝，互不干扰，所以无法统一修改；

　　无法拥有父类原型中定义的方法；

　　子类的实例对象不能通过 instanceof 操作符判断是否是父类的实例。

原型链：指定子类的prototype为父类的一个实例对象。

优缺点和构造函数借用刚好相反。这里特别说明下属性之间相互干扰（对应构造函数借用的优点2）。

组合式继承：上面两种方式互补一下，即用构造方法借用来继承父类属性，用原型链来继承父类方法。

优点：

　　封装保护了内部数据的完整性；

　　封装使对象的重构更轻松；

　　弱化模块间的耦合，提高对象的可重用性；

　　有助于避免命名空间冲突。

缺点：

　　私用方法很难测试；

　　必须与复杂的作用域链打交道，使错误调度更困难；

　　容易形成过度封装；

　　JavaScript并不原生支持封装，所以在JavaScript中实现封装存在复杂性的问题。

## **简述下cookie的操作，还有cookie的属性都知道哪些。**

答：cookie是浏览器提供的一种机制，它将document 对象的cookie属性提供给JavaScript。可以由JavaScript对其进行控制，而并不是JavaScript本身的性质。cookie是 存于用户硬盘的一个文件，这个文件通常对应于一个域名，当浏览器再次访问这个域名时，便使这个cookie可用。因此，cookie可以跨越一个域名下的 多个网页，但不能跨越多个域名使用。可用在保存用户登录状态。跟踪用户行为。定制页面。创建购物车。

$.cookie(‘cookieName’,'cookieValue’，｛expires：7，path：’/'，domain: ‘chuhoo.com’，secure: false，raw:false｝);

注：expires：有效时间；path：设置能够读取cookie的顶级目录；domain： 创建cookie所在网页所拥有的域名；secure：默认是false，如果为true，cookie的传输协议需为https；raw：默认为 false，读取和写入时候自动进行编码和解码（使用encodeURIComponent编码，使用decodeURIComponent解码），关闭 这个功能，请设置为true。

## **IE与FF的JS兼容性都知道哪些。**

答：a.监听事件时：

IE中识别target.attachEvent(type,listener);

FF中识别target.addEventListener(type,listener,useCapture);  
target： 文档节点、document、window 或 XMLHttpRequest。   
type： 字符串，事件名称，FF不含“on”，比如“click”、“mouseover”、“keydown”等。   
listener ：实现了 EventListener 接口或者是 JavaScript 中的函数。   
useCapture ：是否使用捕捉，一般用 false 。

例如：document.getElementById("testText").addEventListener("keydown", function (event) { alert(event.keyCode); }, false);

b.在IE中，可以使用eval("idName")或getElementById("idName")来取得id为idName的HTML对象；[Firefox](http://www.ittang.com/firefox/" \t "http://cache.baiducontent.com/_blank)下只能使用getElementById("idName")来取得id为idName的HTML对象。

c.在 IE 中不能使用 const 关键字。

d.IE可以通过id或者name访问这个frame对应的window对象，而[Firefox](http://www.ittang.com/firefox/" \t "http://cache.baiducontent.com/_blank)只可以通过name来访问这个frame对应的window对象。

## **jquery 闭包 是什么意思 有什么用呢**

jquery中闭包就是函数用到外部变量，不需要传参就可以获取。  
jquery中闭包写法：  
(function($){  
$("div p").click();  
})(jQuery);   
这里面的$只是形参，但jquery是全局变量，所以不需要调用该函数就会自动执行，或者分两步  
  
就是转化成正常的函数，先写函数，后调用。

## **闭包是什么意思？**

闭包是可以包含自由（未绑定到特定对象）变量的代码块；这些变量不是在这个代码块内或者任何全局上下文中定义的，而是在定义代码块的环境中定义。“闭包” 一词来源于以下两者的结合：要执行的代码块（由于自由变量被包含在代码块中，这些自由变量以及它们引用的对象没有被释放）和为自由变量提供绑定的计算环境（作用域）。在 Scala、Scheme、Common Lisp、Smalltalk、Groovy、JavaScript、Ruby 和 Python，objective c 等语言中都能找到对闭包不同程度的支持。

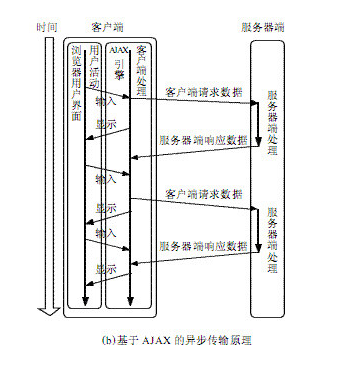
闭包就是能够读取其他函数内部变量的函数。

http://blog.csdn.net/gaoshanwudi/article/details/7355794 此链接可查看（问这个问题的不是一个公司）

## **AJAX是什么? AJAX的交互模型(流程)? AJAX跨域的解决办法?**

答：全称Asynchronous Javascript + XML，是一种用于创建快速动态网页的技术。是一种浏览器和服务器交互技术，基本思想是允许一个浏览器向一个远程页面/服务做异步的http调用，并且用收到的数据来更新一个当前web页面而不必刷新整个页面。该技术能够改进客户端的体验。

交互流程：



跨域问题：

跨域问题简单的理解就是因为JS同源策略的限制（安全性限制），a.com域名下的JS无法操作b.com或c.a.com下的对象。Ajax可以采用jsonp来实现跨域。

JSONP的最基本的原理是：

动态添加一个<script>标签，而script标签的src属性是没有跨域的限制的。这样说来，这种跨域方式其实与ajax XmlHttpRequest协议无关了。当GET请求从被调用页面返回时，可以返回一段JavaScript代码，这段代码会自动调用主页面中的一个callback函数。

Jsonp优点：

不受同源策略的影响，它的兼容性更好，在更加古老的浏览器中都可以运行，不需要XMLHttpRequest或ActiveX的支持；并且在请求完毕后可以通过调用callback的方式回传结果。

Jsonp缺点：

它只支持GET请求而不支持POST等其它类型的HTTP请求；它只支持跨域HTTP请求这种情况，不能解决不同域的两个页面之间如何进行JavaScript调用的问题。

## **正则表达式有系统学习过吗(看书或网上教程)？有的话就问问简单点的邮箱验证、URL验证， 或者问问 贪婪匹配与懒惰匹配 的理论知识。**

**正则表达式**是对字符串操作的一种逻辑公式，就是用事先定义好的一些特定字符、及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。

**邮箱验证：**

/^\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*$/

评注：表单验证时很实用

**URL验证：**

[a-zA-z]+://[^\s]\*

评注：网上流传的版本功能很有限，上面这个基本可以满足需求

当正则表达式中包含能接受重复的限定符时，通常的行为是（在使整个表达式能得到匹配的前提下）**匹配尽可能多的字符**。考虑这个表达式：a.\*b，它将会匹配最长的以a开始，以b结束的字符串。如果用它来搜索aabab的话，它会匹配整个字符串aabab。这被称为**贪婪匹配**。

有时，我们更需要**懒惰匹配**，也就是**匹配尽可能少的字符**。前面给出的限定符都可以被转化为懒惰匹配模式，只要在它后面加上一个问号?。这样.\*?就意味着匹配任意数量的重复，但是在能使整个匹配成功的前提下使用最少的重复

Node.js是否有过尝试？到什么程度？说说个人理解的看法？

基础算法题（如快速排序，能否一两句说说重要的核心原理或者数组消重等）。

是否有接触过或者了解过重构。