

**1、 引入JS的三种方式**

1、在HTML标签中直接使用，直接内嵌JS（但是不提倡使用）；

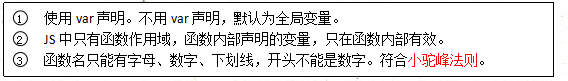
>>>不符合W3C关于内容和行为分离的要求；

2、在HTML页面中使用<scrip> </script>标签包裹JS代码；

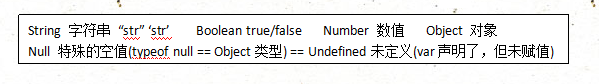
>>>script标签可以放到页面的各种位置；

3、引入外部的JS文件使用<script> </script>标签；

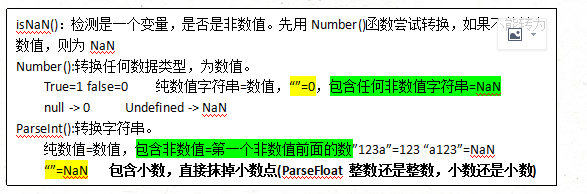
**2、 JS中的变量声明**



**3、 JS中变量的数据类型**

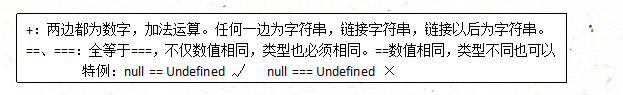


**4、 常用的数值函数**

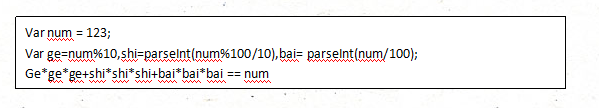


parseInt()只能转字符串，转其他的类型都是NAN；而Number（）能转所有的数据类型；

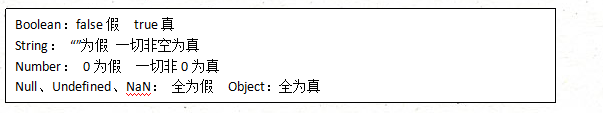
**5、 运算符注意事项**



**6、 判断水仙花数**



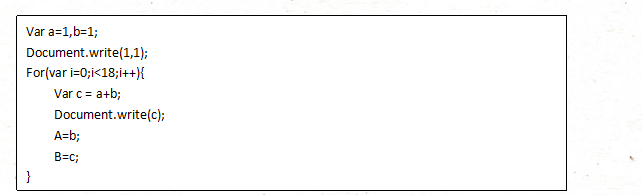
**7、 If结构，循环结构，支持的判断结果：**



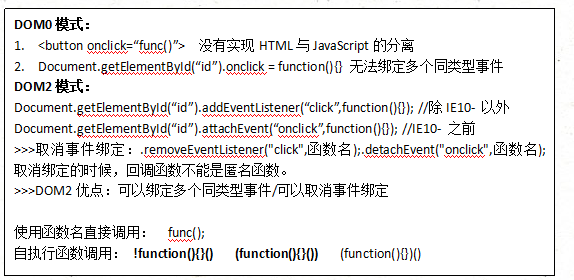
**8、 循环控制语句：**

IMG_263

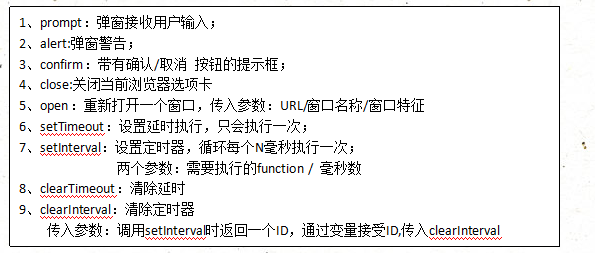
**9、 斐波那契数列:1,1,2,3,5,8,13,21……**



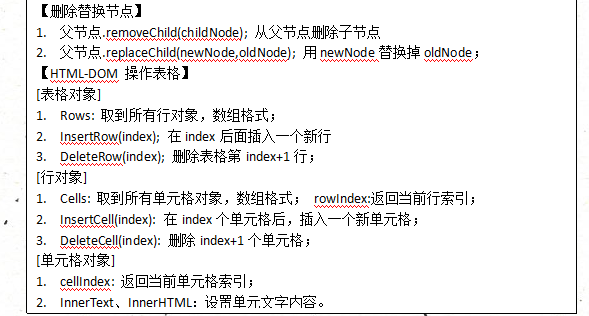
**10、 JS中函数的调用方式/事件的绑定方式**



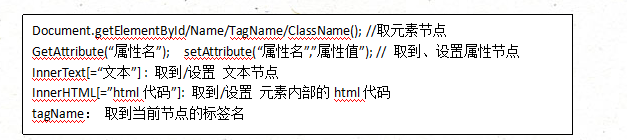
**11、 window对象的常用方法**



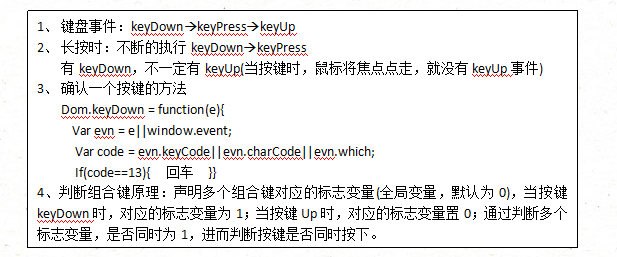




**12、 DOM操作的相关方法**



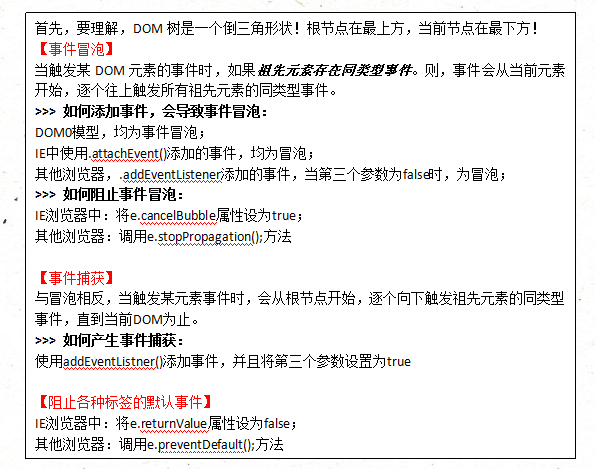
**13、 键盘事件&确定键盘按键**



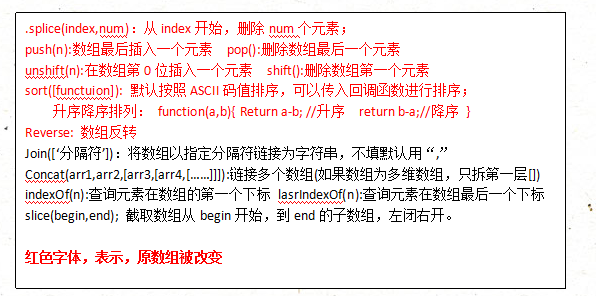
**14、 DOM2 兼容写法：**



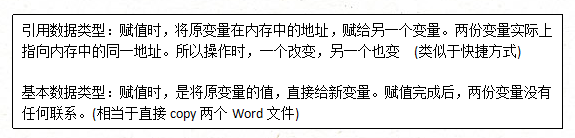
**15、 事件冒泡&事件捕获&阻止默认事件**



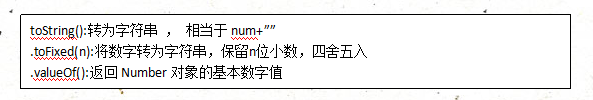
**16、 数组常用方法：**



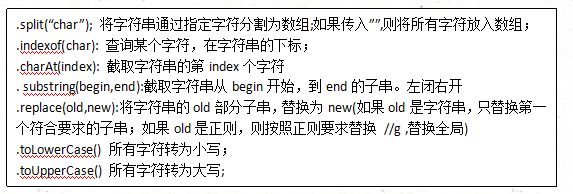
**17、 引用数据类型&基本数据类型：**



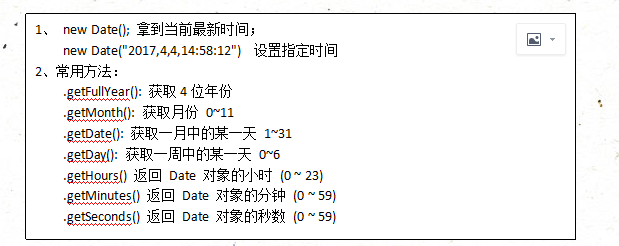
**18、 Number类常用方法：**



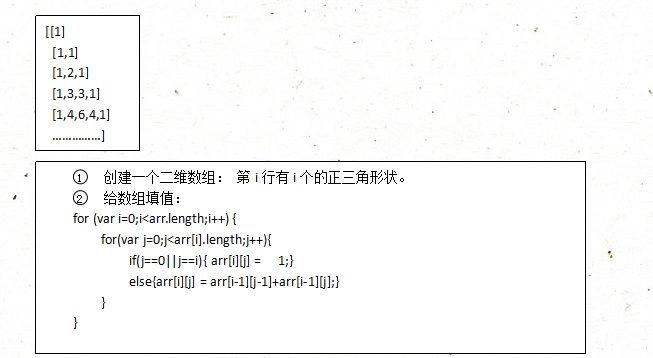
**19、 String类常用方法：**



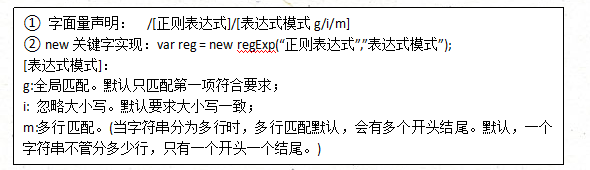
**20、 Date类：**



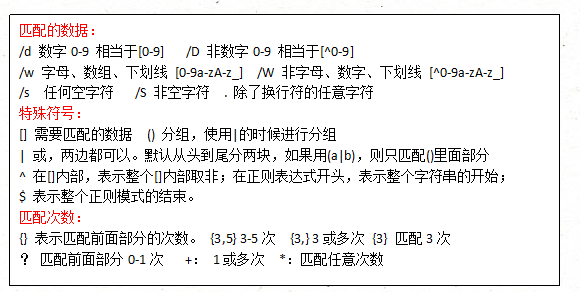
**21、 杨辉三角：**



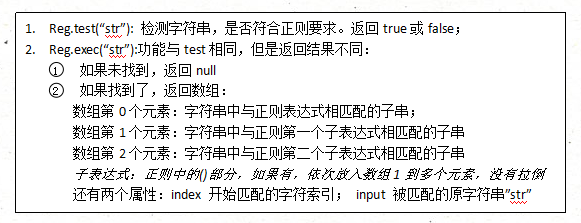
**22、 正则表达式声明：**



**23、 正则表达式常用规则：**



**24、 正则中的test与exec方法：**



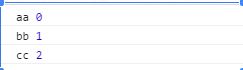
**25、 foreach与each的区别？**

forEach是js中的方法（针对数组），而each是jquery中的方法（针对jquery对象，即$( )；

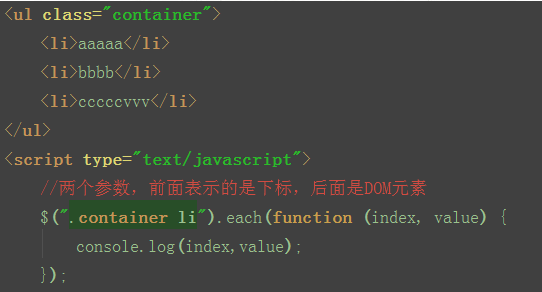
1.forEach方法：如下图所示



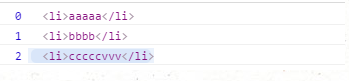
输出结果：



2.each方法：如下图所示



输出结果：



**26、document load 和document ready的区别**

Document.onload 是在结构和样式加载完才执行js

window.onload：不仅仅要在结构和样式加载完，还要执行完所有的样式、图片这些资源文件，全部加载完才会触发window.onload事件

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

**27、什么是CSS Hack？怎么表示？**

CSS hack由于不同厂商的浏览器，比如Internet Explorer,Safari,Mozilla Firefox,Chrome等，或者是同一厂商的浏览器的不同版本，如IE6和IE7，对CSS的解析认识不完全一样，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。

简单的说，CSS hack的目的就是使你的CSS代码兼容不同的浏览器。当然，我们也可以反过来利用CSS hack为不同版本的浏览器定制编写不同的CSS效果。

方式一 条件注释法 方法二 类内属性前缀法 方式三：选择器前缀法

**28、Javascript中callee和caller的作用？**

arguments.callee：获得当前函数的引用

caller是返回一个对函数的引用，该函数调用了当前函数；

callee是返回正在被执行的function函数，也就是所指定的function对象的正文。

**29、简述创建函数的几种方式**

第一种（函数声明）：

function sum1(num1,num2){

return num1+num2;

}

第二种（函数表达式）：

var sum2 = function(num1,num2){

return num1+num2;

}

匿名函数：

function(){}:只能自己执行自己

 第三种（函数对象方式）：

var sum3 = new Function("num1","num2","return num1+num2");

**30、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？**

如果说放在body的封闭之前，将会阻塞其他资源的加载

如果放在body封闭之后，不会影响body内元素的加载

**31、iframe的优缺点？**

优点：

1. 解决加载缓慢的第三方内容如图标和广告等的加载问题

2. Security sandbox安全沙盒

3. 并行加载脚本

缺点：

1. iframe会阻塞主页面的Onload事件

2. 即使内容为空，加载也需要时间

3. 没有语意

**32、请你谈谈Cookie的弊端？**

缺点：

1.`Cookie`数量和长度的限制。

每个domain最多只能有20条cookie，每个cookie长度不能超过4KB，否则会被截掉。

2.安全性问题。

如果cookie被人拦截了，那人就可以取得所有的session信息。即使加密也与事无补，因为拦截者并不需要知道cookie的意义，他只要原样转发cookie就可以达到目的了。

3.有些状态不可能保存在客户端。

例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

**33、js延迟加载的方式有哪些？**

1. defer和async

2. 动态创建DOM方式（创建script，插入到DOM中，加载完毕后callBack）

3. 按需异步载入js

**34、documen.write和 innerHTML 的区别？**

document.write 只能重绘整个页面

innerHTML 可以重绘页面的一部分

**35、哪些操作会造成内存泄漏？**

内存泄漏指任何对象在您不再拥有或需要它之后仍然存在。

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。

1. 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。

2. 闭包

3. 控制台日志

4. 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）

**36、JS是一门什么类型的语言?**

JS可以模拟实现继承和封装，但是不能模拟实现多态，故js是基于事件的，基于对象的轻量级脚本语言。

**37、语言有哪几种类型？各有什么特点？**

1、语言的分类：

①、面向机器：汇编语言。

②、面向过程：C语言

③、面向对象：Java、C++、PHP等。

2、面向过程与面向对象：

①、面向过程：专注于如何去解决一个问题的过程步骤，编程的特点是由一个个的函数去实现每一步的过程步骤，没有类和对象的概念。

②、面向对象：专注于由哪一个对象来解决这个问题，编程特点是出现了一个个的类，从类中拿到对象

【面向对象的三大特征】 继承、封装、多态

【区别】

对于调用者来说，面向过程需要调用者自己去实现各种函数，

而面向对象，只需要告诉调用者对象中具体方法的功能，不需要调用者去了解方法中的实现细节。

**38、JavaScript中类和对象的关系：**

类是抽象的，对象是具体的

类是对象的抽象化，对象是类的具体化；

★类是一个抽象的概念，只能说类有属性和方法，但是不能给属性和方法赋具体的值。

例如:人类有身高，但是不能说出具体数值

★对象是一个具体的个例，是将类中的属性进行具体赋值而来的个体；

例如：张三是人类中的一个个体，身高180cm。

**39、实现继承的三种方式？**

1、使用call bind apply来实现继承。

2、通过扩展object来实现继承。

3、使用原型实现继承。

**40、 什么叫封装？有什么方法和特点？**

① 方法的封装： 将类内部的函数进行私有化处理，不对外提供调用接口，无法在类外部使用的方法，称为私有方法，即方法的封装。

② 属性的封装： 将类中的属性进行私有化处理，对外不能直接使用对象名访问(私有属性)。 同时，需要提供专门用于设置和读取私有属性的set/get方法，让外部使用我们提供的方法，对属性进行操作。 这就叫属性的封装。

封装不是拒绝访问，而是限制访问。 要求调用者，必须使用我们提供的set/get方法进行属性的操作，而不是直接拒绝操作。

因此，单纯的属性私有化，不能称为封装！必须要私有化之后，提供对应的set/get方法。

**41、 原生JS对象与JQuery对象之间的相互转化方式有哪些？**

① 使用$("")选中的是JQuery对象，只能调用JQuery函数，而不能使用元素的JS事件与函数；

② 原生的JS对象转为 JQuery对象

可以使用$() 包裹即可以转为JQuery对象

var p = document.getElementsByTagName("p");

$("p").click(); √ 原生的JS对象已经转为 JQuery对象；

③ JQuery转为原生JS对象：

使用get(index)或者[index]

$("#p").get(0).onclick() = function () {}; √

$("#p").[0].onclick() = function () {}; √

**42、JS常用的内置对象有哪些？它们各自的一些方法有哪些？：**

1、Boolean类、Number类:、String类、Date日期类、

2、常用方法：

**Number类：**

.toString() 将数值类型转换成字符串类型；

.toLocaleString()将数值按照本地格式顺序转换成字符串，一般从右开始，三个一组加都好分隔开；

.toFixed(n)；将数字保留n位小数，并将其转换为字符串格式；

.toPrecision(n):将数字格式化为指定长度。n表示不含小数点的位数长度。如果n<原数字长度，则用科学计数法表示。

.valueOf() 返回Number对象的基本数字值；

**String类：**

**str.length 返回字符串的长度，即为字符数；**

str.toLowerCase()将字符串所有的字符转成小写；

str.UpLowerCase()将字符串所有的字符转成大写；

str.charAt(N)截取数组的第N个字符，相当于str[N]；

str.indexOf("srt",index)查找子串在字符串中出现的位置，如果没有找到返回-1，其他同数组的indexOf方法；

str.lastIndexOf()同数组。

str.subString (begin,end) 截取数组中的某一部分并返回截取的字符串

>>>传入一个参数，表示是开始的区间，默认将截取到字符串末尾；

>>>传入两个参数，表示开始和结束的下标，是一个左闭右开区间；

str.split(分隔符)将字符串以指定分隔符分隔存入数组中，

str.replace(old，new)将字符串中的字符进行替换；

**Date日期类：**

1、new Date();返回当前最新时间；

new Date("2017,08,29,12:33:29") 返回指定时间；

**43、什么是Jquery中的文档就绪函数？**

文档就绪函数防止文档在未完全加载之前，运行JQuery代码；

$(document).ready(function() {

//JQuery代码

})；

简写形式如下：

$(function(){});

[文档就绪函数&window.onload的区别]

① window.onload必须等到网页中的所有内容(包括图片视频)加载完成后才能执行代码；

文档就绪函数只需要在 网页DOM结构加载完成之后，就可以执行代码；

② window.onload只能写一个，写多个之后只会执行最后一个；

文档就绪函数可以写多个，而且不会覆盖。

**44、常用的CSS的预处理器有哪些？**

less

sass 常用的编译方式

它们之间的区别：

SCSS中的嵌套：选择器嵌套 属性嵌套 伪类嵌套

混合宏、继承、占位符：

① 混合宏:使用@mixin声明混合宏，在其他选择器中使用@include调用混合宏

② 继承 ：声明一个普通的class，在其他的选择器中使用extend 继承这个class；

③ 占位符 ：使用%声明%占位符，在其他选择器中使用@extend 继承占位符；

**45、简述下JS中的事件分类**

1、鼠标事件： click/dblclick/onmouseon/onmouseover.....

2、HTML事件： onload/onscroll/onsubmit/onchange/onfocus....

3、键盘事件：

keydown:键盘按下时触发

keypress:键盘按住时触发

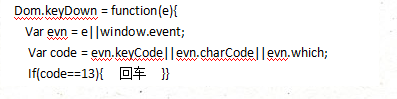
keyup:键盘抬起时触发

【确定触发的按键】

① 在触发的函数中传入一个参数e，表示键盘事件；

② 使用e.keyCode，取到按键的Ascii码值，进而确定触发按键；

③ 所有浏览器的兼容写法(一般并不需要)



**46、浅谈JavaScript的优点和缺点：**

**优点：**

* 性能：由于JavaScript运行在客户端，节省了web服务器的请求时间和带宽
* 轻量级的脚本语言，比较容易学习
* 运行在用户机器上，运行结果和处理相对比较快。
* 可以使用第三方附加组件来检查代码片段。

**缺点：**

* 安全问题：由于JavaScript在客户端运行，可能被用于黑客目的。
* 渲染问题：在不同浏览器中的处理结果可能不同。

**47、浅谈jQuery的优点和缺点：**

使用jQuery最大的好处是少量的代码做更多的事情。

Write Less Do More

看个简单例子，获取元素的值，使用JavaScript：

Javascript代码

1. document.getElementById(‘elementid').value

使用jQuery：

Jquery代码

1. $(‘#elementid').val();

与JavaScript相比，jQuery的语法更加简单。**通过jQuery，可以很容易地浏览文档、选择元素、处理事件以及添加效果等，同时还允许开发者定制插件。**

**优点：**

* jQuery消除了JavaScript跨平台兼容问题。
* 相比其他JavaScript和JavaScript库，jQuery更容易使用。
* jQuery有一个庞大的库/函数。
* jQuery有良好的文档和帮助手册。
* jQuery支持AJAX。

**缺点：**

* 由于不是原生JavaScript语言，理解起来可能会受到限制。
* 项目中需要包含jQuery库文件。如果包含多个版本的jQuery库，可能会发生冲突。

**48、浅谈angular优缺点**

简介：angularjs是一个javascript框架。通过script脚本引入，他是一个用Javascript编写的库。**angularjs通过指令扩展了HTML，通过表达式绑定数据到HTML中。AngularJS主要考虑的是构建CRUD（增删改查）应用。幸运的是，至少90%的WEB应用都是CRUD应用。**

**优点：**

1. 改变操作dom的方式

将传统的JQuery的先选择元素，在操作的方式改变成了直接对于元素本身的操作。

这依赖于强大的Html Parser的能力和directive灵活。

2. 后端MVC前端化

3. 数据操作

**缺点：**

1、框架略笨重

2、不适合页面复杂的网站类型开发：

内容网站，需要SEO的。(SEO目前也有了prerender解决方案) https//[prerender.io](https://link.zhihu.com/?target=http://prerender.io/)

交互频繁的，如游戏之类交互体验网站。

**49、什么叫做事件委派？有什么作用？（优点）**

Jq中的事件委派on

>>>将原本需要绑定到子元素上的事件，绑定到其祖先节点乃至根节点上面，再委派给子元素节点，生效！

eg: $("p").on("click",function(){});

事件委派 如下：

$(document).on("click","p",function(){});

>>>作用：

不使用事件委派的绑定方式，只能将事件绑定到初始化时的子元素标签上，

当页面新增同类型标签时，这些新增的标签上，没有之前绑定的事件。

而使用 事件委派时 ，当页面新增更同类型标签时,这些新增的事件也具有之同类型前标签所绑定的事件！

off()取消事件绑定

$("p").off("click"):取消单事件绑定

$("p").off("click mouseover dbclick"):取消多事件绑定

$(document).off("click"，"p"):取消委派事件绑定

$("p").off()取消所有的事件绑定

**50、详细介绍下 CSS的动画**

**1、 css的transition transition抓住了所设置变化属性的起始态和完成态，通过设定的速度曲线来完成动画。可以涉及到各种变化的css属性，默认为all，则所有变化的属性都会在出发时，以动画的形式展现出来。**

tips：transform是一种变化属性，该属性允许我们对元素进行旋转、缩放、移动或倾斜。可以作为transition中需要变化的属性。

**2.css3的animation属性**

使用animation属性制作动画可以更加灵活的设置动画帧，通过不同keyframe（动画帧）的设置，实现很多优雅的效果，keyframe中的百分数是动画完成总时间的比例。

animation是设置总的动画效果，而keyframe中设置上相应的动画名字，然后在keyframe中设置具体的动画效果。

JS中的动画

**3.原生js动画**

原生js动画利用js代码，将动画一步以函数的方式写出来，可以实现多种动画样式，而且可以自己做兼容性处理，自己设立每一步的动画效果，并且能够完成比较复杂的效果，但是代码量很大。

51、**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

**52、jquery 中如何将数组转化为json字符串，然后再转化回来？**

jQuery中没有提供这个功能，所以你需要先编写两个jQuery的扩展：

[IMG_287](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

$.fn.stringifyArray = function(array) {

return JSON.stringify(array)

}

$.fn.parseArray = function(array) {

return JSON.parse(array)

}

然后调用：

$("").stringifyArray(array)

[IMG_288](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

**53、函数实现阻止冒泡函数的功能？**

|  |  |
| --- | --- |
|  | function stopPropagation(e) {      e = e || window.event;      if(e.stopPropagation) { //W3C阻止冒泡方法          e.stopPropagation();      } else {          e.cancelBubble = true; //IE阻止冒泡方法      }  }  document.getElementById('need\_hide').onclick = function(e) {      stopPropagation(e);  } |

**54、什么是闭包？ 写一个简单的闭包？**

  我的理解是，闭包就是能够读取其他函数内部变量的函数。在本质上，闭包就是将函数内部和函数外部连接起来的一座桥梁。

JS中提供了一种闭包的概念：在函数中，定义一个子函数，子函数可以访问父函数的私有变量，

可以在子函数中进行操作，最后将子函数通过return返回。

闭包的作用:

① 可以在函数外部访问函数的私有变量；

② 让函数内部的变量，可以始终存在于内存中，不会在函数调用完成之后立即释放！

|  |  |
| --- | --- |
|  | function outer(){      var num = 1;      function inner(){          var n = 2;          alert(n + num);      }      return inner;  }  outer()(); |

**55、call和apply的区别是什么？**

    参数形式不同，call(obj, pra, pra)后面是单个参数。apply(obj, [args])后面是数组。

**56、为什么不能定义1px左右的div容器？**

**IE6下这个问题是因为默认的行高造成的，解决的方法也有很多，例如：**

overflow:hidden | zoom:0.08 | line-height:1px

**57、bind(), live(), delegate()的区别**

    bind： 绑定事件，对新添加的事件不起作用，方法用于将一个处理程序附加到每个匹配元素的事件上并返回jQuery对象。

    live： 方法将一个事件处理程序附加到与当前选择器匹配的所有元素（包含现有的或将来添加的）的指定事件上并返回jQuery对象。

delegate： 方法基于一组特定的根元素将处理程序附加到匹配选择器的所有元素（现有的或将来的）的一个或多个事件上。

**58、简述link和import的区别？**

区别1：link是XHTML标签，除了加载CSS外，还可以定义RSS等其他事务；@import属于CSS范畴，只能加载CSS。

区别2：link引用CSS时，在页面载入时同时加载；@import需要页面网页完全载入以后加载。

区别3：link是XHTML标签，无兼容问题；@import是在CSS2.1提出的，低版本的浏览器不支持。

区别4：link支持使用Javascript控制DOM去改变样式；而@import不支持。

**59、你如何优化自己的代码？**

1、代码重用

2、避免全局变量（命名空间，封闭空间，模块化mvc..）

3、拆分函数避免函数过于臃肿：**单一职责原则**

**4、适当的注释，尤其是一些复杂的业务逻辑或者是计算逻辑，都应该写出这个业务逻辑的具体过程**

内存管理，尤其是闭包中的变量释放

**60、使用js实现这样的效果：在文本域里输入文字时，当按下enter键时不换行，而是替换成“{{enter}}”,(只需要考虑在行尾按下enter键的情况).**

textarea.onkeydown=function(e){

  e.preventDefault();//为了阻止enter键的默认换行效果

  if(e.keycode==”enter键码”){

    testarea.value+=”{{enter}}”;

  }

**61、请简要描述web前端性能需要考虑哪方面，你的优化思路是什么？**

参见雅虎14web优化规则

减少http请求：

1、小图弄成大图，

2、合理的设置缓存

3、资源合并、压缩

将外部的js文件置底

**62、简述readyonly与disabled的区别**

readonly只针对input(text / password)和textarea有效，

而disabled对于所有的表单元素都有效，当表单元素在使用了disabled后，当我们将表单以POST或GET的方式提交的话，这个元素的值不会被传递出去，而readonly会将该值传递出去

**63、请尽可能详尽的解释ajax的工作原理**

**思路：先解释异步，再解释ajax如何使用**

Ajax的原理简单来说通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用javascript来操作DOM而更新页面。这其中最关键的一步就是从服务器获得请求数据。要清楚这个过程和原理，我们必须对 XMLHttpRequest有所了解。

　XMLHttpRequest是ajax的核心机制，它是在IE5中首先引入的，是一种支持异步请求的技术。简单的说，也就是javascript可以及时向服务器提出请求和处理响应，而不阻塞用户。达到无刷新的效果。

**64、为什么扩展javascript内置对象不是好的做法？**

因为扩展内置对象会影响整个程序中所使用到的该内置对象的原型属性

**65、浏览器标准模式和怪异模式之间的区别是什么？**

标准模式是指，浏览器按W3C标准解析执行代码；

怪异模式则是使用浏览器自己的方式解析执行代码，因为不同浏览器解析执行的方式不一样，所以我们称之为怪异模式。

浏览器解析时到底使用标准模式还是怪异模式，与你网页中的DTD声明直接相关，DTD声明定义了标准文档的类型（标准模式解析）文档类型，会使浏览器使用相应的方式加载网页并显示，忽略DTD声明,将使网页进入怪异模式。

**66、如果设计中使用了非标准的字体，你该如何去实现？**

先通过font-face定义字体，再引用

@font-face

{

font-family: myFirstFont;

src: url('Sansation\_Light.ttf'),

     url('Sansation\_Light.eot'); /\* IE9+ \*/

}

**67、简述.trigger与.triggerHandler的区别**

.trigger()自动 触发某元素事件；

第一个参数：是需要触发的事件类型；

第二个参数：(可选)数组格式，表示传递给事件函数的参数；

>>>传递进来的参数，可以在事件函数中，定义形参进行获取

(形参第一个必须是event事件，所以从第二个开始为所传递的参数！)

>>> 也可以直接在函数中，使用arguements对象数组，读取参数;

 .triggerHandler()：功能同上，区别如下：

① .triggerHandler()不能触发浏览器默认的HTML事件，如submit等；

② .trigger()可以触发页面中所有匹配元素的事件；

而.triggerHandler()只能触发第一个匹配元素的事件；

③ .trigger()的返回值，返回的是调用当前函数的对象，符合JQuery的可链式语法；

.triggerHandler()返回的是事件函数的返回值，如果事件函数没有返回值，则返回的是Undefined；

**69、call bind apply 这三个函数有什么共同的作用？**

通过函数名调用这三个函数，可以强行将函数中的this指定为某一个对象

三个函数的唯一区别，在于接受func的参数列表的形式不同，除此之外，功能上没有差别！

三个函数的写法(区别)

① call写法：func.call(func的this指向的obj,func参数1,func参2,....);

② apply写法：func.apply(func的this指向的obj,[func参数1,func参2,....]);

③ bind写法：func.bind(func的this指向的obj),(func参数1,func参2,....);

**70、扩展Object实现继承**

① 声明一个父类函数

function Parent (){}

声明一个子类函数

function Son (){}

② 通过prototype给Object类添加一个扩展方法：

Object.prototype.extend = function (parent) {

for(var i in parent){

this[i] = parent[i];

}

}

③ 分别拿到父类对象和子类对象: 实例化

var p = new Parent();

var s = Son();

④ 用子类对象调用扩展方法，实现从继承操作：

s.extend(p)

实现继承的原理：

**通过循环将父类对象的所有属性和方法，全部赋给子类对象。**

关键点在于for-in循环，即使不扩展Object，也能通过简单的循环实现操作；

**71、详细解释下JS中的this指向问题**

谁最终调用函数，this就指向谁！

this指向的规律 ：(跟函数的调用方式息息相关！)

① 通过 【 函数名() 】 调用的，this永远指向window；

② 通过 【 对象.方法 】调用的，this永远指向对象。obj.func();

③ 函数作为数组中的 一个元素，通过数组下标调用的 【 arr[i]() 】，this指向数组arr。

④ 函数作为window内置函数的回调函数使用，this指向widow；

setTimeout() setInterval()等

⑤ 函数作为构造函数，使用new关键字调用，this指向新new出的对象。

**72、BootStrap的优点和缺点:**

优点：

**流媒体网格布局**

**响应式设计**

**自定义表单元素**

**页面排版**

**JavaScript交互性**

**跨浏览器兼容性**

BT栅格系统的本质就是通过CSS3的媒体查询实现的

BootStrap的缺点:

兼容性可能不太好

**73、什么是渐进增强（progressive enhancement）、优雅降级（graceful degradation）呢？**

渐进增强（progressive enhancement）：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。（从被所有浏览器支持的基本功能开始，逐步地添加那些只有新式浏览器才支持的功能，向页面添加无害于基础浏览器的额外样式和功能。当浏览器支持时，它们会自动地呈现出来并发挥作用。）

优雅降级（graceful degradation）：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。（Web站点在所有新式浏览器中都能正常工作，如果用户使用的是老式浏览器，则代码会检查以确认它们是否能正常工作。由于IE独特的盒模型布局问题，针对不同版本的IE的hack实践过优雅降级了，为那些无法支持功能的浏览器增加候选方案，使之在旧式浏览器上以某种形式降级体验却不至于完全失效。）

区别：优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的、能够起作用的版本开始，并不断扩充，以适应未来环境的需要。

**74、常见的浏览器状态码**

200--请求成功！

500--内部服务器错误！

404--请求失败！

**75、详述jQuery中Ajax的写法以及get和post的区别**

$.ajax() 通过 HTTP 请求加载远程数据。jQuery 最底层 AJAX 实现函数。

接受一个大的对象。对象里面的属性和方法，表示ajax请求的相关设置：

① url:被请求的远程文件的路径。

② Ajax请求的类型，可选值get/post.

回调函数：

•success： 请求成功的回调函数。接受一个参数data，表示后台返回的数据。

•dataType：预期服务器返回的数据类型。

"json": 返回 JSON 数据 。

"text": 返回纯文本字符串

•data：对象格式。向后台发送一个对象，表示传递的数据。

常用于type为post的请求方式。

如果type为get,可以使用？直接追加载URL后面。

•error:请求失败时的回调函数

•statusCode:接受一个对象，对象的键值对是status状态码和对应的回调函数，

表示当请求状态码是对应数字时，执行具体的操作函数.

$.get(); $.post(); 这两个函数，是在$.ajax()的基础上进行封装而来。可以直接默认发送get请求或post请求；

\*

\* 接受四个参数：

\* ① 请求的URL路径。 相当于$.ajax()里面的url；

\* ② 向后台传递的数据。 相当于$.ajax()里面的data；

\* ③ 请求成功的回调函数。 相当于$.ajax()里面的success；

\* ① 预期返回的数据类型。 相当于$.ajax()里面的dataType；

**76、IE和DOM事件流的区别**

**IE9以前：attachEvent(“onclick”)、detachEvent(“onclick”)**

**IE9开始跟DOM事件流是一样的，都是addEventListener**

**77、IE跟DOM事件流有什么区别**

① 阐述事件绑定方式：

**IE9以前：attachEvent(“onclick”)、detachEvent(“onclick”)**

**IE9开始:跟DOM事件流是一样的，都是addEventListener**

② 这两种绑定方式的区别：

   写法上的区别：

attachEvent两个参数、事件带on

addEventListener 三个参数、事件不带on、true/false表示捕获/冒泡

功能上的区别：

attachEvent绑定的事件全部冒泡、addEventListener根据第三个参数是true还是false决定冒泡还是捕获。

③ 事件冒泡和事件捕获。

   冒泡： 当触发一个事件时，会从当前节点开始，依次触发其祖先节点的同类型事件。

   捕获： 从根节点开始，依次触发其祖先节点同类型事件，直到节点自身。

**78、IE和标准下有哪些兼容性的写法？**

1、获取事件对象(事件因子)：  ev = e || window.event;

2、DOM2添加事件： if(dom.attachEvent){

Dom. attachEvent(“onclick”,function);

}else{

Dom. addEventListener(“click”,function);

}

3、取消事件冒泡： if (e.stopPropagation) {

         e.stopPropagation(); //IE以外

    } else {

        e.cancelBubble = true; //IE8之前

   }

4、取消默认事件：if (e.preventDefault) {

     e.preventDefault(); //IE以外

  } else {

     e.returnValue = false; //IE

  }

**79、API是什么？**

API（Application Programming Interface,应用程序编程接口）是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

在编程语言中，API通常指语言中内置的函数、接口、类等系统工具。我们编程人员无需关心这些函数的实现细节，只需要按照API文档的要求，给函数传入指定参数接受返回值即可。

在前后台传递数据，API也是后台提供给前台的接口，前台只需要按照要求请求接口并发送指定参数，结课接受JSON字符串。例如：<https://api.douban.com/v2/book/1220562>

**80、Javascript事件处理器在线程空闲之前不会运行是什么意思？**

JS是一个单线程应用，也就是说，当某一段代码正在执行的时候，其他代码如果需要执行也必须等到这个线程结束后，才能执行。

|  |  |
| --- | --- |
|  | for(var i=1;i<=3;i++){    setTimeout(function(){        console.log(i);    },0);  }; |

代码执行到for循环时，当for循环执行第一次，遇到第一个setTimeout时并不会立即触发，因为for循环还有结束，现在的线程被for循环阻塞。setTimeout必须等到for循环的线程结束以后，也就是线程空闲之后，才能执行，而这时候i已经变成了4.

    所以最终打印三个4.  解决的办法可以将var改成let，或者for循环中嵌套自执行函数。 原理。。。。。

**81、js中callee和caller的作用**

写法不同：

callee是Arguments的一个属性，在函数中使用Arguments.callee调用。

Caller在函数中，直接使用函数名调用。  Func. Caller

功能不同：

Arguments.callee 返回的是，当前函数自身的引用！！

Func. Caller 返回的是，当前函数在哪个函数中调用。 如果函数时顶层调用，则返回null

function func(){

console.log(func.caller);

}

func(); // null

function func1(){

func();

}

func1(); // 返回func1

**82、js中的保留字是什么？**

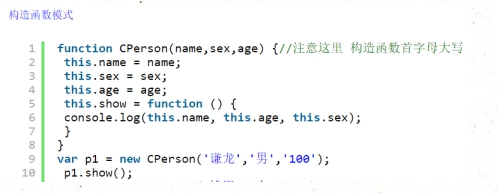
程序中预定义的关键字(函数名、类名、属性名、方法名等标识符)都是js中的保留字。 这些保留字，不允许用户再次声明为变量、方法、函数

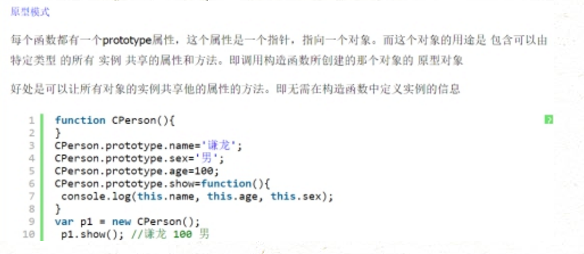
保留字在编译器中会变色。

**83、工厂方式创建js对象是啥方式 （JS中创建对象的多种方式）**

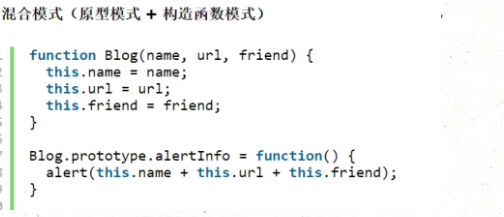
 ① 字面量模式： var obj = {}







混合模式就是我们所说的将属性添加到成员属性(构造函数模式)，将方法添加到原型方法（原型模式）



**84、js延迟加载的方式有哪些？**

<http://blog.csdn.net/dragoo1/article/details/48155501>

+将JS代码放在文档最后也是延迟加载的方式之一。

**85、documen.write和 innerHTML 的区别？**

documen.write 是直接在整个文档中，写入代码。会覆盖掉其他已有代码。

innerHTML 是选中一个节点后，修改节点里面的代码，只影响当前节点。

**86、解释jsonp的原理，以及为什么不是真正的ajax**

后续讲解

**87、javascript 中的垃圾回收机制？闭包中的内存释放？**

Js中的垃圾回收机制，函数中的变量，会在函数执行完成后，被回收处理。

但是，如果函数中使用了闭包，那么函数中的变量将始终被内部的函数持有，而不会在函数执行完成后释放。

<http://www.cnblogs.com/zhwl/p/4664604.html>

闭包的两个重要作用：

① 可以让函数外部，能够访问函数内部的变量。

② 让函数内部的变量在函数执行完成后，不会被释放，而始终保持在内存中

**88、哪些操作会造成内存泄漏**

大量使用全局变量

大量使用闭包

清除DOM节点时，只清除节点，而没有删除事件

<http://www.cnblogs.com/libin-1/p/6013490.html>

**89、js中的精度问题**

JS中在浮点数运算或者大整数运算的时候，可能导致不准确的结果出现。 例如0.7+0.1=0.7999999999999 解决办法可以将数字\*10相加，然后再除以10

[参考;http://www.cnblogs.com/snandy/p/4943138.html](http://www.cnblogs.com/snandy/p/4943138.html)

**90、简述defer和async的区别**

**1.默认引用 script:<script type="text/javascript" src="x.min.js"></script>**

　　当浏览器遇到 script 标签时，**文档的解析将停止，并立即下载并执行脚本，脚本执行完毕后将继续解析文档。**

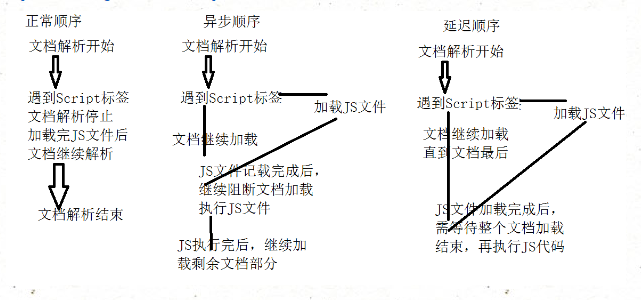
**2.async模式 <script type="text/javascript" src="x.min.js" async="async"></script>**

　　当浏览器遇到 script 标签时，文档的**解析不会停止，其他线程将下载脚本，脚本下载完成后开始执行脚本，脚本执行的过程中文档将停止解析，直到脚本执行完毕。**

**3.defer模式 <script type="text/javascript" src="x.min.js" defer="defer"></script>**

　　当浏览器遇到 script 标签时，文档的解析不会停止，其他线程将下载脚本，**待到文档解析完成，脚本才会执行。**

<http://www.cnblogs.com/xuechenlei/p/5947555.html>



**91、业界常用的优化WEB页面加载速度的方法（可以分别从页面元素展现，请求连接，css,js,服务器等方面介绍）**

[参考： http://www.cnblogs.com/MarcoHan/p/5295398.html](http://www.cnblogs.com/MarcoHan/p/5295398.html)

**92、解释什么是sql注入，xss漏洞？**

 参考:   <http://www.cnblogs.com/ITtangtang/p/3982297.html>

**93、解释下这个css选择器发生什么？   [role=nav]>ul a:not([href^=mailto]){}**

[role=nav] 选中页面中 role属性等于nav的元素. 也就是导航栏

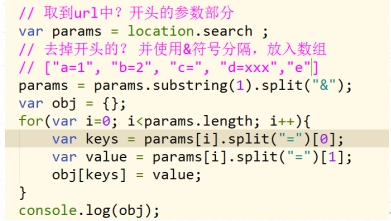
[role=nav]>ul 导航栏中的子节点 ul

[role=nav]>ul a  UL里面的a标签

a:not([href^=mailto]) 选中a标签，除了href属性使用mailto开头的。

选中导航栏中的ul里面的不是mailto的a标签。

**94、有这样一个URL：http://item.taobao.com/item.htm?a=1&b=2&c= e &d=xxx，请写一段JS程序提取URL中的各个GET参数(参数名和参数个数不确定)，将其按key-value形式返回到一个json结构中，如{a:’1′,?b:’2′,?c:”,?d:’xxx’,?e:undefined}。**



**95、以下哪个单词不属于javascript保留字   B**

A.with

B.parent

C.class

D.void

**96、 动态创建DOM方式（创建script，插入到DOM中，加载完毕后callBack）**

**97、 按需异步载入js**

[参考：http://www.jb51.net/article/34491.htm](http://www.jb51.net/article/34491.htm)

**99、简单的介绍下JS面向对象中的\_\_proto\_\_与prototype**

1、prototype:函数的原型对象

① 只有函数才有prototype，而且所有的函数必然有prototype!

② prototype本身也是一个对象！

③ prototype指向了当前函数所在的引用地址！

2、\_\_proto\_\_：对象的原型

① 只有对象才有\_\_proto\_\_，而且所有的对象必有\_\_proto\_\_；

② \_\_proto\_\_也是一个对象，所以也有自己的\_\_proto\_\_，顺着这条线向上找的顺序，就是原型链。

③ 数组都是对象，也都有自己的\_\_proto\_\_；

3、实例化一个类，拿到对象的原理：

实例化一个类的时候，实际上是将新对象的\_\_proto\_\_，指向构造函数所在的prototype；

也就是说：zhangsan.\_\_proto\_\_ ==Person.prototype √

4、所有对象的\_\_proto\_\_沿着原型链向上查找都将指向Object的prototype;

Object的prototype的原型，指向null；

**100、简述下什么是CSShack以及其表现形式有哪些？**

CSS hack由于不同厂商的浏览器，比如Internet Explorer,Safari,Mozilla Firefox,Chrome等，或者是同一厂商的浏览器的不同版本，如IE6和IE7，对CSS的解析认识不完全一样，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。

简单的来说：

CSS hack的目的就是使你的CSS代码兼容不同的浏览器。

表现形式：

（1）方式一 条件注释法

（2）方式二 类内属性前缀法

（3）CSS hack方式三：选择器前缀法

[http://note.youdao.com/share/?id=f396eab5bbadbca8c30acb68f9ec6ded&type=note#/](http://note.youdao.com/share/?id=f396eab5bbadbca8c30acb68f9ec6ded&type=note" \l "/)

**101、Require.js有什么作用？**

1、实现了文件的异步加载，避免网页失去响应。

2、管理模块之间的依赖性，便于代码的编写和维护。

**102、怎么检测一个对象是不是一个类的实例？**

instanceof：A instanceof B 检测一个对象(A)是不是一个类(B)的一个实例；

louver instanceof Person;√ louver是函数Person的实例化；

louver instanceof Object;√ 所有对象都是Object的一个实例；

Person instanceof Object;√ 函数本身也是一个对象；

constructor：返回当前对象的构造函数；

**103、zepto和jQuery的不同之处有哪些？**

1、zepto不支持IE浏览器；

2、zepto增加了一些基本的触摸事件：如tap，swip事件’

3、DOM操作的区别：添加ID时jQuery不会生效，而zepto会生效！

4、使用zepto时，load事件的处理函数会执行，使用jQuery时，load事件的处理函数不会执行。

5、事件委派的不同：

在zepto中，当a被点击之后，一次性弹出内容为a事件和b事件，但是在jQuery中只会触发委托弹出a事件，再点击一次才会触发b事件！

6、offse:zepto：返回left top width heigth

jQuery:只会返回left top

7、zepto无法隐藏元素的宽高，jQuery可以；

8、zepto不支持的选择器：

① 基本伪类：first last eq(index)

② 内容伪类：contains(text) :empty :has(selector)

③ 可见性伪类：hidden visible

④ 属性选择器：[attribute = value]

⑤ 表单伪类：input :text :password :submit

⑥ 表单对象属性：selected

**104、使用JS跳转页面的方法：**

Window.location="http://www.baidu.com";

细分几种使用location跳转页面的类型：

location.replace("http://www.baidu.com"); 使用新的文档替换当前文档，替换以后，不能退回。

location.assign("http://www.baidu.com");加载新的文档，加载之后，可以退回。

location.repload();重新加载当前页面，即刷新页面；

①、 location.repload(true) 强制刷新页面，从服务器端重新加载页面----Ctrl+f5;

②、 location.repload() 在本地刷新当前页面---f5

**105、一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验。**

图片懒加载，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。

如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。

如果图片为css图片，可以使用CSSsprite，SVGsprite，Iconfont、Base64等技术。

如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。

如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致。

更换最新的图片格式 谷歌的webp腾讯的tpg

合成雪碧图

大图分割加载

**106、CSS3新增伪类有那些？**

 p:first-of-type 选择属于其父元素的首个 <p> 元素的每个 <p> 元素。

    p:last-of-type  选择属于其父元素的最后 <p> 元素的每个 <p> 元素。

    p:only-of-type  选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。

    p:only-child    选择属于其父元素的唯一子元素的每个 <p> 元素。

    p:nth-child(2)  选择属于其父元素的第二个子元素的每个 <p> 元素。

    :enabled、:disabled 控制表单控件的禁用状态。

:checked，单选框或复选框被选中。

**107、如何在 HTML5 页面中嵌入音频、视频?**

HTML 5 包含嵌入音频文件的标准方式，支持的格式包括 MP3、Wav 和 Ogg：

<audio controls>

  <source src="jamshed.mp3" type="audio/mpeg">

   Your browser does'nt support audio embedding feature.

</audio>

和音频一样，HTML5 定义了嵌入视频的标准方法，支持的格式包括：MP4、WebM 和 Ogg：

<video width="500px" height="500px" autoplay="autoplay" loop="loop">

<source src="binzhiliang1.mp4" type="video/mp4" >

Your browser does'nt support video embedding feature.

</source>

</video>

**108、怎么取到一个网址并转化为字符串格式存储？**

console.log(location.href);//返回当前完整路径；

/\*将js中的对象、数组，转化\*/

var str1 = JSON.stringify(jsonObj);

console.log(str1);

var obj = JSON.parse(str1);

console.log(obj);

**109、如何优化网页加载速度？**

   1.减少css，js文件数量及大小(减少重复性代码，代码重复利用)，压缩CSS和Js代码

   2.图片的大小

   3.把css样式表放置顶部，把js放置页面底部

   4.减少http请求数

   5.使用外部 Js 和 CSS

**110、知道css有个content属性吗？有什么作用？有什么应用？**

知道。css的content属性专门应用在 before/after 伪元素上，用来插入生成内容。最常见的应用是利用伪类清除浮动。

//一种常见利用伪类清除浮动的代码

.clearfix:after {

    content:"."; //这里利用到了content属性

    display:block;

    height:0;

    visibility:hidden;

    clear:both; }

.clearfix {

    \*zoom:1;

}

after伪元素通过 content 在元素的后面生成了内容为一个点的块级素，再利用clear:both清除浮动。

**111、简述下web storage和cookie的区别**

1、Web Storage相对于cookie来说，拥有更大的容量储存。

2、Cookie的大小是受限的，并且每次你请求一个新的页面的时候Cookie都会被发送过去，这样无形 中浪费了带宽。

3、cookie还需要指定作用域，不可以跨域调用。

4、Web Storage拥有setItem,getItem,removeItem,clear等方法，不像cookie需要前端开发者自己封装setCookie，getCookie。

5、但是Cookie也是不可以或缺的：Cookie的作用是与服务器进行交互，作为HTTP规范的一部分而存在 ，而Web Storage仅仅是为了在本地“存储”数据而生。

**112、为什么利用多个域名来存储网站资源会更有效？**

CDN缓存更方便

突破浏览器并发限制

节约cookie带宽

节约主域名的连接数，优化页面响应速度

防止不必要的安全问题

**113、HTML5 Canvas 元素有什么用？**

答：Canvas 元素用于在网页上绘制图形，该元素标签强大之处在于可以直接在 HTML 上进行图形操作。

**115、请用js实现随机选取10–100之间的10个数字，存入一个数组，并排序。**

[IMG_295](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

var iArray = [];

funtion getRandom(istart, iend){var iChoice = istart - iend +1;return Math.floor(Math.random() \* iChoice + istart;

}for(var i=0; i<10; i++){

iArray.push(getRandom(10,100));

}

iArray.sort();

[IMG_296](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

**116、如何消除一个数组里面重复的元素？**

[IMG_297](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

方法一：var arr=[1,2,3,3,4,4,5,5,6,1,9,3,25,4];function deRepeat(){var newArr=[];var obj={};var index=0;var l=arr.length;for(var i=0;i<l;i++){if(obj[arr[i]]==undefined){

obj[arr[i]]=1;

newArr[index++]=arr[i];

}else if(obj[arr[i]]==1)

}return newArr;

}var newArr2=deRepeat(arr);

alert(newArr2); //输出1,2,3,4,5,6,9,25方法二：var arr = [1,2,2,3,3,3,4,5,6];function func (arr){var newArr = new Array;

newArr.push(arr[0]);for (var i=0;i<=arr.length;i++) {if (newArr.indexOf(arr[i])==-1) {

newArr.push(arr[i]);

}

}

document.write(newArr);

}

func(arr);

[IMG_298](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

**117、闭包：下面这个ul，如何点击每一列的时候alert其index？**

[IMG_299](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

<ul id="test">

<li>这是第一条</li>

<li>这是第二条</li>

<li>这是第三条</li>

</ul>非闭包实现var lis=document.querySelectorAll('li');

document.querySelector('#test').onclick=function(e){

for (var i = 0; i < lis.length; i++) {

var li = lis[i];

if(li==e.target){

alert(i);

}

}

};//闭包实现var lis=document.querySelectorAll('li');for (var i = 0; i < lis.length; i++) {

var li = lis[i];

li.onclick=(function(index){

return function(e){

alert(index);

};

})(i);

}

[IMG_300](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

**118、JavaScript数组元素添加、删除、排序等方法有哪些？**

Array.concat( ) 连接数组

Array.join( ) 将数组元素连接起来以构建一个字符串

Array.length 数组的大小

Array.pop( ) 删除并返回数组的最后一个元素

Array.push( ) 给数组添加元素

Array.reverse( ) 颠倒数组中元素的顺序

Array.shift( ) 将元素移出数组

Array.slice( ) 返回数组的一部分

Array.sort( ) 对数组元素进行排序

Array.splice( ) 插入、删除或替换数组的元素

Array.toLocaleString( ) 把数组转换成局部字符串

Array.toString( ) 将数组转换成一个字符串

**119、简述remove和detach异同**

\* 1、相同点：

\* ① 都会把当前标签，以及当前标签的所有子节点，全部删除；

\* ② 都可以在删除时，把当前节点返回。并可以使用变量接受返回的节点，以便后期恢复；

\* 2、 不同点：

\* 使用接受的节点，恢复原节点时。

\* remove只能恢复节点的内容，但是事件绑定，不能再恢复；

\* detach不但恢复节点的内容，还能再恢复 事件的绑定；

**120、Ajax 是什么? 如何创建一个Ajax？**

Ajax并不算是一种新的技术，全称是asynchronous javascript and xml，可以说是已有技术的组合，主要用来**实现客户端与服务器端的异步通信效果，实现页面的局部刷新，带来更好的用户体验；按需获取数据，节约带宽资源；早期的浏览器并不能原生支持ajax，可以使用隐藏帧（iframe）方式变相实现异步效果，后来的浏览器提供了对ajax的原生支持**

使用ajax原生方式发送请求主要通过XMLHttpRequest(标准浏览器)、ActiveXObject(IE浏览器)对象实现异步通信效果

**121、举例简述同步和异步的区别?**

**同步：阻塞的**

-张三叫李四去吃饭，李四一直忙得不停，张三一直等着，直到李四忙完两个人一块去吃饭

=浏览器向服务器请求数据，服务器比较忙，浏览器一直等着（页面白屏），直到服务器返回数据，浏览器才能显示页面

**异步：非阻塞的**

-张三叫李四去吃饭，李四在忙，张三说了一声然后自己就去吃饭了，李四忙完后自己去吃

=浏览器向服务器请求数据，服务器比较忙，浏览器可以自如的干原来的事情（显示页面），服务器返回数据的时候通知浏览器一声，浏览器把返回的数据再渲染到页面，局部更新

**122、如何解决跨域问题?**

理解跨域的概念：协议、域名、端口都相同才同域，否则都是跨域

出于安全考虑，服务器不允许ajax跨域获取数据，但是可以跨域获取文件内容，所以基于这一点，可以动态创建script标签，使用标签的src属性访问js文件的形式获取js脚本，并且这个js脚本中的内容是**函数调用，该函数调用的参数是服务器返回的数据，为了获取这里的参数数据，需要事先在页面中定义回调函数，在回调函数中处理服务器返回的数据，这就是解决跨域问题的主流解决方案**

**123、页面编码和被请求的资源编码如果不一致如何处理？**

对于ajax请求传递的参数，如果是get请求方式，参数如果传递中文，在有些浏览器会乱码，不同的浏览器对参数编码的处理方式不同，所以对于get请求的参数需要使用 encodeURIComponent函数对参数进行编码处理，后台开发语言都有相应的解码api。对于post请求不需要进行编码

**124、简述ajax 的过程。**

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

写法如下：

$.ajax() 通过 HTTP 请求加载远程数据。jQuery 最底层 AJAX 实现函数。

接受一个大的对象。对象里面的属性和方法，表示ajax请求的相关设置：

① url:被请求的远程文件的路径。

② Ajax请求的类型，可选值get/post.

回调函数：

•success： 请求成功的回调函数。接受一个参数data，表示后台返回的数据。

•dataType：预期服务器返回的数据类型。

"json": 返回 JSON 数据 。

"text": 返回纯文本字符串

•data：对象格式。向后台发送一个对象，表示传递的数据。

常用于type为post的请求方式。

如果type为get,可以使用？直接追加载URL后面。

•error:请求失败时的回调函数

•statusCode:接受一个对象，对象的键值对是status状态码和对应的回调函数，

表示当请求状态码是对应数字时，执行具体的操作函数.

404 - 页面没有找到！

403 - 服务器收到请求，但是拒绝提供服务！

200 - 请求成功！

500 - 内部服务器错误！

503 - 服务器当前不能处理客户端的请求，一段时间后可能恢复正常

301 - 资源（网页等）被永久转移到其它URL！

**125、阐述一下异步加载。**

1. 异步加载的方案： 动态插入 script 标签

2. 通过 ajax 去获取 js 代码，然后通过 eval 执行

3. script 标签上添加 defer 或者 async 属性

4. 创建并插入 iframe，让它异步执行 js

**126、请解释一下 JavaScript 的同源策略。**

所谓同源指的是：协议，域名，端口相同，同源策略是一种安全协议，指一段脚本只能读取来自同一来源的窗口和文档的属性。

在前后台进行数据交互时，两个文件必须处于同一域名同一端口号同一协议名之下，否则跨域请求

**127、ajax的缺点**

 1、ajax不支持浏览器back按钮。

 2、安全问题 AJAX暴露了与服务器交互的细节。

 3、对搜索引擎的支持比较弱。

 4、破坏了程序的异常机制。

**128、解释jsonp的原理，以及为什么不是真正的ajax**

　　Jsonp并不是一种数据格式，而json是一种数据格式，**jsonp是用来解决跨域获取数据的一种解决方案，具体是通过动态创建script标签，然后通过标签的src属性获取js文件中的js脚本，该脚本的内容是一个函数调用，参数就是服务器返回的数据，为了处理这些返回的数据，需要事先在页面定义好回调函数，本质上使用的并不是ajax技术**

**129、什么是Ajax和JSON，它们的优缺点。**

Ajax是全称是asynchronous JavaScript andXML，即异步JavaScript和xml，**用于在Web页面中实现异步数据交互，实现页面局部刷新。**

优点：可以使得页面不重载全部内容的情况下加载局部内容，降低数据传输量，避免用户不断刷新或者跳转页面，提高用户体验

缺点：对搜索引擎不友好；要实现ajax下的前后退功能成本较大；可能造成请求数据的增加跨域问题限制；

**JSON是一种轻量级的数据交换格式，ECMA的一个子集**

优点：轻量级、易于人的阅读和编写，便于机器（JavaScript）解析，支持复合数据类型（数组、对象、字符串、数字）

**130、分别介绍下javascript的本地对象，内置对象和宿主对象**

本地对象为独立于宿主环境的ECMAScript提供的对象，包括Array Object RegExp等**可以new实例化的对象**

内置对象为Gload，Math 等**不可以实例化的(他们也是本地对象，内置对象是本地对象的一个子集)**

宿主对象为所有的非本地对象，所有的BOM和DOM对象都是宿主对象，如浏览器自带的document,window 等对象。

图解：



**131、为什么利用多个域名来存储网站资源会更有效？**

1、速度保证：

确保用户在不同地区能用最快的速度打开网站。

2、安全保障：

其中某个域名崩溃用户也能通过其他域名问网站，

3、服务端：

并且不同的资源放到不同的服务器上有利于减轻单台服务器的压力。

**132、请说出三种减低页面加载时间的方法**

1、压缩css、js文件

2、合并js、css文件，减少http请求

3、外部js、css文件放在最底下

4、减少dom操作，尽可能用变量替代不必要的dom操作

5、优化图片加载的方式。

**133、 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?**

Webkit是浏览器引擎，包括html渲染和js解析功能，手机浏览器的主流内核，与之相对应的引擎有Gecko（Mozilla Firefox 等使用）和Trident（也称MSHTML，IE 使用）。

对于浏览器的调试工具要熟练使用，主要是页面结构分析，后台请求信息查看，js调试工具使用，熟练使用这些工具可以快速提高解决问题的效率.

IMG_302

1、Elements 功能标签页：用来查看，修改页面上的元素，包括DOM标签，以及css样式的查看，修改，还有相关盒模型的图形信息

2、Console控制台：用于打印和输出相关的命令信息，常用于JS错误调试。

3、Sources js资源页面：这个页面内我们可以找到当然浏览器页面中的js 源文件，方便我们查看和调试(打断点调试)

4、.Network 网络请求标签页：可以看到所有的资源请求，包括网络请求，图片资源，html,css，js文件等请求，可以根据需求筛选请求项，一般多用于网络请求的查看和分析，分析后端接口是否正确传输，获取的数据是否准确，请求头，请求参数的查看

5、.Timeline标签页可以显示JS执行时间、页面元素渲染时间，不做过多介绍

6、.Profiles标签页可以查看CPU执行时间与内存占用，不做过多介绍

7、.Resources标签页会列出所有的资源，以及HTML5的Database和LocalStore等，你可以对存储的内容编辑和删除 不做过多介绍

8、.Security标签页 可以告诉你这个网站的安全性，查看有效的证书等

9、.Audits标签页 可以帮你分析页面性能，有助于优化前端页面，分析后得到的报告

**134、简述一下 Handlebars 的基本用法？**

Handlebars 是 JavaScript 一个语义模板库，通过对view和data的分离来快速构建Web模板。

**135、详解在AngualrJS中使用$http进行数据请求的方法？**

[IMG_303](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

$http({

method: 'GET', /\*请求的方法\*/

url: 'h51701.json' /\*请求的地址\*/

}).then(function(obj){/\*请求成功的回调函数\*/

alert("请求成功！");

$scope.data = obj.data;

},function(){/\*请求失败的回调函数\*/

alert("请求失败！");

});

[简写方式]

可以直接简写为get或者post方式：

$http.get('/someUrl').then(successCallback, errorCallback);

$http.post('/someUrl', data).then(successCallback, errorCallback);

[IMG_304](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

**136、如何让一个标签块级显示？**

1、给该元素添加浮动属性

2、给该元素添加绝对定位属性

3、 给父元素添加display:flex属性设置为弹性盒子（弹性盒子内部全是块级标签）

4、给其本身添加display:block/ inlineblock属性

**137、利用原生JS实现Ajax请求后台数据的步骤？**

[IMG_305](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

// 第一步： 获得XMLHttpRequest对象var ajax = new XMLHttpRequest();// 第二步： 设置状态监听函数

ajax.onreadystatechange = function(){

console.log(ajax.readyState);

console.log(ajax.status);

// 第五步：在监听函数中，判断readyState=4 && status=200表示请求成功if(ajax.readyState==4 && ajax.status==200){// 第六步： 使用responseText、responseXML接受响应数据，并使用原生JS操作DOM进行显示console.log(ajax.responseText);

console.log(ajax.responseXML);// 返回不是XML，显示nullconsole.log(JSON.parse(ajax.responseText));

console.log(eval("("+ajax.responseText+")"));

}

}

// 第三步： open一个链接

ajax.open("GET","h51701.json",false);//true异步请求，false同步

// 第四步： send一个请求。 可以发送对象和字符串，不需要传递数据发送null

ajax.send(null);

[IMG_306](http://www.cnblogs.com/wq1994/p/javascript:void(0);)

**138、实现Ajax的跨域请求有哪些方法？**

【跨域请求处理】

由于JS中存在同源策略，当请求不同协议名，不同端口号，不同主机名下面的文件时，将会违背同源策略，无法请求成功！需要进行跨域处理！

1、后台PHP进行设置，在后台请求的PHP文件中，写入一条header:

header( "Access-Control-Allow-Origin:\*");

表示允许哪些域名请求这个PHP文件，\*表示所有的域名都允许。

2、使用src属性+JSONP实现跨域

① 拥有src属性的标签自带跨域功能，所以可以使用<script>标签的src属性请求后台属性。

<script type="text/javascript" src="http://127.0.0.1/json.php"> </ script>

② 由于src在加载数据成功之后，会直接将加载内容放入到script标签中，

所以后台直接返回JSON字符串将不能在script标签中解析。

因此，后台应该返回给前台一个回调函数名，并将JSON字符串作为参数传入，

后台PHP文件中返回 ： echo "callBack($str)";

③ 前台接收到返回的回调函数，将直接在script标签中调用。因此需要声明这样的一个回调函数，作为请求成功的回调函数。

\* function callBack(data) {

alert("请求成功！");

}

3、jQuery的Ajax实现JSONP

① 在Ajax请求时设置dataType:"jsonp"

② 后台返回时，依然需要返回回调函数，但是，在ajax发送请求时，会默认使用get请求将回调函数名发给后台，

后台可以使用$\_GET['callback']取出回到函数名。

echo "{$\_GET['callback']}({$str})";

③ 后台返回之后，Ajax依然可以使用succes作为成功的回调函数：

success:function(data){

console.log(data);

}

当然后台也可以随便返回一个回调函数名： echo "callBack($json)";

前台只要请求成功就会自动调用这个函数。类似于上面的第2条的② ③ 步；

**139、js中捕获异常是啥？**

Js中的异常捕获用**try{}catch(){} 结构进行捕获。**

将程序中可能出现异常错误的代码，用try块包裹，当代码真正出现异常时，程序流将进入carch块捕获异常，try块中从异常语句开始下面的代码将不再执行，转而执行catch块中的代码！！！！

JS中常见的异常类型：

EvalError: raised when an error occurs executing code in eval()

RangeError: raised when a numeric variable or parameter is outside of its valid range

ReferenceError: raised when de-referencing an invalid reference

SyntaxError: raised when a syntax error occurs while parsing code in eval()

TypeError: raised when a variable or parameter is not a valid type

URIError: raised when encodeURI() or decodeURI() are passed invalid parameters

  所有异常都继承自Error类

用户可以自定义异常： throw new Error() ;   throw new TypeError(“错误信息”);

Catch块的小括号中可以接受异常对象e, e.name表示错误名，e.message表示错误信息。

**140、事件委派的原理是啥？事件冒泡吗？**

事件委派的原理就是事件冒泡！

当你点击一个元素时，这个点击事件是会冒泡的，事件依次从该元素向父级元素依次触发，事件委托就是在父级元素上统一监听和处理事件，而不是在具体的元素上，然后通过判断事件源来确定是哪个元素触发的。

这样做的好处是：

我们不需要为每个元素添加事件，更节省资源；并且后来新增的元素直接可以同样获得事件，不需要再为其添加监听。

**141、前端代码优化的方法？**

**1.  [避免坏请求](http://www.feedthebot.com/pagespeed/avoid-bad-requests.html)**

**2.  [避免CSS @import](http://www.feedthebot.com/pagespeed/avoid-css-import.html)**

**3.  [避免使用document.write](http://www.feedthebot.com/pagespeed/avoid-document-write.html)**

**4.  [合并多个外部CSS文件](http://www.feedthebot.com/pagespeed/combine-external-css.html)**

**5.  [合并多个外部JavaScript文件](http://www.feedthebot.com/pagespeed/combine-external-javascript.html)**

**6.  [通过CSS sprites来整合图像](http://www.feedthebot.com/pagespeed/combine-images-css-sprites.html)**

**7. [延迟JavaScript的加载](http://www.feedthebot.com/pagespeed/defer-loading-javascript.html)**

**8.  [将小的CSS和JavaScript代码内嵌到HTML中](http://www.feedthebot.com/pagespeed/inline-small-css.html)**

**9、  [利用浏览器缓存](http://www.feedthebot.com/pagespeed/leverage-browser-caching.html)**

**142、详细解释一下关于伪数组与数组？（在Javascript中什么是伪数组？如何将伪数组转化为标准数组？）**

伪数组：

具有length属性；

按索引方式存储数据；

不具有数组的push()、pop()等方法；

var fakeArray01 = {0:'a',1:'b',length:2};//这是一个标准的伪数组对象

var arr01 = Array.prototype.slice.call(fakeArray01);

<http://www.cnblogs.com/momo798/p/6024633.html>

**143、Javascript作用域链?**

函数作用域的嵌套就组成了所谓的函数作用域链

<http://www.cnblogs.com/renlong0602/p/4398883.html>

**144、什么是JSOPN？**

jsonp：JSON with padding，是JSON的 一种“使用模式”，可用于解决主流浏览器的跨域数据访问的问题。

**145、如何利用Ajxa实现跨域请求？**

由于JS中存在同源策略，当请求不同协议名，不同端口号，不同主机名下面的文件时，将会违背同源策略，无法请求成功！需要进行跨域处理！

1、后台PHP进行设置，在后台请求的PHP文件中，写入一条header:

header( "Access-Control-Allow-Origin:\*");

表示允许哪些域名请求这个PHP文件，\*表示所有的域名都允许。

2、使用src属性+JSONP实现跨域

① 拥有src属性的标签自带跨域功能，所以可以使用<script>标签的src属性请求后台属性。

\* <script type="text/javascript" src="http://127.0.0.1/json.php"> </ script>

② 由于src在加载数据成功之后，会直接将加载内容放入到script标签中，

\* 所以后台直接返回JSON字符串将不能在script标签中解析。

\* 因此，后台应该返回给前台一个回调函数名，并将JSON字符串作为参数传入，

后台PHP文件中返回 ： echo "callBack($str)";

③ 前台接收到返回的回调函数，将直接在script标签中调用。因此需要声明这样的一个回调函数，作为请求成功的回调函数。

\* function callBack(data) {

alert("请求成功！");

}

3、利用jQuery的Ajax实现JSONP

① 在Ajax请求时设置dataType:"jsonp"

② 后台返回时，依然需要返回回调函数，但是，在ajax发送请求时，会默认使用get请求将回调函数名发给后台，

后台可以使用$\_GET['callback']取出回到函数名。

echo "{$\_GET['callback']}({$str})";

③ 后台返回之后，Ajax依然可以使用succes作为成功的回调函数：

success:function(data){

console.log(data);

}

当然后台也可以随便返回一个回调函数名： echo "callBack($json)";

前台只要请求成功就会自动调用这个函数。类似于上面的第2条的② ③ 步；

**146、 JavaScript window.onload 事件和 jQuery ready 函数有何不同？**

**1.执行时间**

    window.onload必须等到页面内包括图片的所有元素加载完毕后才能执行。

    $(document).ready()是DOM结构绘制完毕后就执行，不必等到加载完毕。

**2.编写个数不同**

     window.onload不能同时编写多个，如果有多个window.onload方法，只会执行一个

     $(document).ready()可以同时编写多个，并且都可以得到执行

**3.简化写法**

     window.onload没有简化写法

     $(document).ready(function(){})可以简写成$(function(){});

**147、$(this) 和 this 关键字在 jQuery 中有何不同?**

1、this表示javascript的dom对象

2、$(this)表示把dom对象转换为jquery对象

3、jquery包的方法只能由jquery对象调用，不能用javascript的Dom对象使用

$("input").click(function(){

//这里的this表示被点击的输入框对象,可以使用javascript对该对象进行操作

var value=this.value;

//jquery对象与dom对象取值方法不一样，要用jquery方法必须使用$(dom对象)转换

var val=$(this).val();

）}

**148、简单介绍下JavaScript中的作用域与变量声明提升？**

1、一个变量的作用域表示这个变量存在的上下文。一个变量的作用域表示这个变量存在的上下文。javascript有拥有函数级别的作用域，也就是说，在一个函数内定义的变量只能在函数内部访问或者这个函数内部的函数访问（闭包除外）每一个在函数外部声明或者定义的变量都是一个全局对象，所以这个变量可以在任何地方被使用。

2、变量声明提升就是在函数内部不用var 声明的一个变量，想要访问的时候，会在当前函数中寻找，发现当前函数中没有，继续往上层函数找，一直到全局作用下还没有找到，就会将这个变量在全局范围内进行声明。

**149、jquery.extend 与 jquery.fn.extend的区别？**

jQuery为开发插件提拱了两个方法，分别是：

jQuery.fn.extend();

jQuery.extend();

jquery.extend：用于声明全局插件/方法。声明插件/方法使用$.func();直接调用

query.fn.extend：用于声明局部插件/方法。声明插件/方法使用$(“选择器”).func();

**150、常见的浏览器内核有哪些？**

1，使用Trident的是internet explorer，国产的绝大部分浏览器。Trident是就是ie内核

2，使用Gecko的是Mozilla Firefox，使用 Gecko 内核的浏览器也有不少，如 Netscape MozillaSuite/SeaMonkey 等

3，使用Presto的是opera，这是目前公认网页浏览速度最快的浏览器内核

4，使用WebKit的有苹果的safari，谷歌的chrome，还有国产的大部分双核浏览器其中一核就是WebKit

**151、margin和padding分别适合什么场景使用？**

margin是用来隔开元素与元素的间距；padding是用来隔开元素与内容的间隔。 margin用于布局分开元素使元素与元素互不相干； padding用于元素与内容之间的间隔，让内容（文字）与（包裹）元素之间有一段

**152、::before 和 :after中双冒号和单冒号 有什么区别？解释一下这2个伪元素的作用。**

单冒号(:)用于CSS3伪类，双冒号(::)用于CSS3伪元素。（伪元素由双冒号和伪元素名称组成）

双冒号是在当前规范中引入的，用于区分伪类和伪元素。不过浏览器需要同时支持旧的已经存在的伪元素写法，

比如:first-line、:first-letter、:before、:after等，

而新的在CSS3中引入的伪元素则不允许再支持旧的单冒号的写法。 想让插入的内容出现在其它内容前，使用::before，否者，使用::after；

在代码顺序上，::after生成的内容也比::before生成的内容靠后。

如果按堆栈视角，::after生成的内容会在::before生成的内容之上

**153、 li与li之间（display:inline-block）有看不见的空白间隔是什么原因引起的？有什么解决办法？**

浏览器的默认行为是把inline元素间的空白字符（空格换行tab）渲染成一个空格，

**154、解释一下JavaScript中的this指针、闭包、作用域？**

This指向：JS函数中的this，永远指向函数调用语句所在的对象(谁调用函数，this指向谁)。(2分)

闭包： 在函数中，写一个子函数。内层函数可以访问外层函数中的变量。(2分)

作用域： 变量生效的范围称为作用域。JS中只有全局作用域和函数作用域。(2分)

**155、使用jQuery操作元素的属性attr()方法和val()有什么区别?**

Attr():用于读取或者设置元素的属性；

Val(): 用于读取或者设置表单元素的value；

**156、JavaScript怎样创建节点、删除节点、替换节点、复制节点？**

创建节点： .creatElement();

删除节点： .removeChild();

替换节点： .replaceChild();

复制节点： .cloneNode(true/false);

**157、CSS中的link和@import有什么区别？常用哪种？**

① link属于标准的HTML标签，而@import不是标准标签；

② link可以兼容所有低版本浏览器，而@import只在CSS2之后能用；

③ link是将两个文件链接起来，起桥梁作用； 而@import相当于将CSS文件复制到HTML文件中；

④ link会在HTML文件边加载的过程中，边链接CSS文件；

@import会在HTML文件全部加载完以后，再导入CSS文件；