**21、如何判断一个对象是否属于某个类？**

  使用instanceof

  if(a instanceof Person){

      alert('yes');

   }

**22、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);

**23、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。

**24、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"}

JSON字符串转换为JSON对象:

var obj =eval('('+ str +')');

var obj = str.parseJSON();

var obj = JSON.parse(str);

JSON对象转换为JSON字符串：

var last=obj.toJSONString();

var last=JSON.stringify(obj);

[].forEach.call($$("\*"),function(a){a.style.outline="1pxsolid #"+(~~(Math.random()\*(1<<24))).toString(16)}) 能解释一下这段代码的意思吗？

**25、js延迟加载的方式有哪些？**

defer和async、动态创建DOM方式（用得最多）、按需异步载入js

**26、Ajax 是什么? 如何创建一个Ajax？**

ajax的全称：AsynchronousJavascript 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

**27、Ajax 解决浏览器缓存问题？**

（1）在ajax发送请求前加上anyAjaxObj.setRequestHeader("If-Modified-Since","0")。

（2）在ajax发送请求前加上anyAjaxObj.setRequestHeader("Cache-Control","no-cache")。

（3）在URL后面加上一个随机数： "fresh=" + Math.random();。

（4）在URL后面加上时间搓："nowtime=" + new Date().getTime();。

（5）如果是使用jQuery，直接这样就可以了 $.ajaxSetup({cache:false})。这样页面的所有ajax都会执行这条语句就是不需要保存缓存记录。

**28、同步和异步的区别?**

同步的概念应该是来自于OS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。

**29、如何解决跨域问题?**

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

**30、模块化开发怎么做？**

立即执行函数,不暴露私有成员

   var module1 = (function(){

    var\_count = 0;

    var m1 =function(){

    //...

    };

    var m2 =function(){

    //...

    };

    return {

    m1 :m1,

    m2 : m2

    };

    })();

**31、AMD（Modules/Asynchronous-Definition）、CMD（CommonModule Definition）规范区别？**

Asynchronous Module Definition，异步模块定义，所有的模块将被异步加载，模块加载不影响后面语句运行。所有依赖某些模块的语句均放置在回调函数中。

 区别：

    （1） 对于依赖的模块，AMD 是提前执行，CMD 是延迟执行。不过RequireJS 从 2.0 开始，也改成可以延迟执行（根据写法不同，处理方式不同）。CMD 推崇 as lazy as possible.

    （2）CMD 推崇依赖就近，AMD 推崇依赖前置。看代码：

// CMD

define(function(require, exports, module) {

   var a = require('./a')

   a.doSomething()

   // 此处略去 100 行

   var b = require('./b') // 依赖可以就近书写

   b.doSomething()

   // ...

})

// AMD 默认推荐

define(['./a', './b'], function(a, b) { // 依赖必须一开始就写好

   a.doSomething()

   // 此处略去 100 行

   b.doSomething()

   // ...

})

**32、异步加载JS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

**33、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

**34、DOM操作——怎样添加、移除、移动、复制、创建和查找节点?**

（1）创建新节点

 createDocumentFragment()    //创建一个DOM片段

 createElement()   //创建一个具体的元素

 createTextNode()   //创建一个文本节点

（2）添加、移除、替换、插入

 appendChild()

 removeChild()

 replaceChild()

 insertBefore() //在已有的子节点前插入一个新的子节点

（3）查找

 getElementsByTagName()    //通过标签名称

 getElementsByName()    //通过元素的Name属性的值(IE容错能力较强，会得到一个数组，其中包括id等于name值的)

 getElementById()    //通过元素Id，唯一性

**35、.call() 和 .apply() 的区别？**

  例子中用 add 来替换 sub，add.call(sub,3,1)== add(3,1) ，所以运行结果为：alert(4);

  注意：js 中的函数其实是对象，函数名是对 Function 对象的引用。

   function add(a,b)

    {

       alert(a+b);

    }

   function sub(a,b)

    {

       alert(a-b);

    }

   add.call(sub,3,1);

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

\* jquery.extend 为jquery类添加类方法，可以理解为添加静态方法

\* jquery.fn.extend:

    源码中jquery.fn= jquery.prototype，所以对jquery.fn的扩展，就是为jquery类添加成员函数

使用：

jquery.extend扩展，需要通过jquery类来调用，而jquery.fn.extend扩展，所有jquery实例都可以直接调用。

**37、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

**38、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：

   $.fn.stringifyArray = function(array) {

       return JSON.stringify(array)

    }

   $.fn.parseArray = function(array) {

       return JSON.parse(array)

    }

    然后调用：

   $("").stringifyArray(array)

**39、针对 jQuery 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：varstr=$("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++) {}

**40、如何判断当前脚本运行在浏览器还是node环境中？（阿里）**

this === window ? 'browser' : 'node';

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

**41、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?**

jquery stop(): 如：$("#div").stop().animate({width:"100px"},100);

**42、那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。

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）

**43、JQuery一个对象可以同时绑定多个事件，这是如何实现的？**

\* 多个事件同一个函数：

   $("div").on("click mouseover", function(){});

\* 多个事件不同函数

   $("div").on({

       click: function(){},

       mouseover: function(){}

});

**44、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浏览器内核。

**45、用js实现千位分隔符?**

function commafy(num) {

   return num && num

       .toString()

       .replace(/(\d)(?=(\d{3})+\.)/g, function($0, $1) {

           return $1 + ",";

       });

}

console.log(commafy(1234567.90));//1,234,567.90

**46、检测浏览器版本版本有哪些方式？**

功能检测、userAgent特征检测

比如：navigator.userAgent

//"Mozilla/5.0 (Macintosh; Intel MacOS X 10\_10\_2) AppleWebKit/537.36

 (KHTML, like Gecko) Chrome/41.0.2272.101 Safari/537.36"

**47、What is a Polyfill?**

polyfill 是“在旧版浏览器上复制标准 API 的 JavaScript 补充”,可以动态地加载JavaScript 代码或库，在不支持这些标准 API 的浏览器中模拟它们。

例如，geolocation（地理位置）polyfill 可以在 navigator 对象上添加全局的 geolocation 对象，还能添加 getCurrentPosition 函数以及“坐标”回调对象，

所有这些都是 W3C 地理位置 API 定义的对象和函数。因为 polyfill 模拟标准 API，所以能够以一种面向所有浏览器未来的方式针对这些 API 进行开发，

一旦对这些 API 的支持变成绝对大多数，则可以方便地去掉 polyfill，无需做任何额外工作。

**48、做的项目中，有没有用过或自己实现一些polyfill 方案（兼容性处理方案）？**

比如： html5shiv、Geolocation、Placeholder

**49、使用JS实现获取文件扩展名？**

function getFileExtension(filename) {

 return filename.slice((filename.lastIndexOf(".") - 1>>> 0) + 2);

}

String.lastIndexOf() 方法返回指定值（本例中的'.'）在调用该方法的字符串中最后出现的位置，如果没找到则返回 -1。

对于'filename'和'.hiddenfile'，lastIndexOf的返回值分别为0和-1无符号右移操作符(»>) 将-1转换为4294967295，将-2转换为4294967294，这个方法可以保证边缘情况时文件名不变。

String.prototype.slice() 从上面计算的索引处提取文件的扩展名。如果索引比文件名的长度大，结果为""。