1. javascript的typeof返回那些数据类型？

答：string 、number 、boolean 、object 、undefind 、function

1. 如何判断某变量是否为数组数据类型？

答：一、判断其是否具有‘数组性质’，如push（）方法。由于自己也可以给该变量定义push方法，所以此方法并不准确。

二、obj instanceof Array在某些IE版本中不正确。

三、方法一二都有漏洞，在ES5中定义了新方法Array.isArray(),保证了其兼容性，最好的方法如下：

If(typeof Array.isArray ==’undefined’){

Array.isArray = function(arg){

Return Object.prototype.toString.call(arg) === “[object Array]”;

}

}

3.例举3种强制类型转换和2种隐式类型转换？

答：强制转换为：parseInt 、 parseFloat 、Number

隐式转换为：== 、- 、\* 、/ 、+ 、> 、< 、>= 、<=

4.split()和join()的区别？

答：split是将某一对象中的某一切割符进行切割，最后返回一个数组。

join是将数组拼接成一个字符串。

5.数组方法pop(),push(),unshift(),shift()

答：push是从尾部添加

pop是从尾部删除

unshift是从头部添加

shift是从头部删除

6.IE和标准下有哪些兼容写法

答：var e = ev || window.event

Document.body || document.documentElement

xxx.currentStyle || window.getComputedStyle(xxx,null)

event.cancelBubble = true || event.stopPropagation()

7．ajax请求的时候get和post两种方式的区别？

答：1、 get是把参数数据队列加到提交表单的ACTION属性所指的URL中，值和表单内各个字段一一对应，在URL中可以看到。post是通过HTTP post机制，将表单内各个字段与其内容放置在HTML HEADER内一起传送到ACTION属性所指的URL地址。用户看不到这个过程。   
2、 对于get方式，[服务器端](https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuWTvuARvuj6dPjn4n1wb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHbsnWT4nHbk)用Request.QueryString获取变量的值，对于post方式，[服务器端](https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuWTvuARvuj6dPjn4n1wb0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHbsnWT4nHbk)用Request.Form获取提交的数据。两种方式的参数都可以用Request来获得。   
3、get传送的数据量较小，不能大于2KB。post传送的数据量较大，一般被默认为不受限制。但理论上，因服务器的不同而异. get安全性非常低，post安全性较高。   
5、 <form method="get" action="a.asp?b=b">跟<form method="get" action="a.asp">是一样的，也就是说，action页面后边带的参数列表会被忽视；而<form method="post" action="a.asp?b=b">跟<form method="post" action="a.asp">是不一样的。

另外   
 Get请求有如下特性：它会将数据添加到URL中，通过这种方式传递到服务器，通常利用一个问号？代表URL地址的结尾与数据参数的开端，后面的参数每一个数据参数以“名称=值”的形式出现，参数与参数之间利用一个连接符&来区分。   
Post请求有如下特性：数据是放在HTTP主体中的，其组织方式不只一种，有&连接方式，也有分割符方式，可隐藏参数，传递大批数据，比较方便。   
   
总而言之：当我们在提交表单的时候我们通常用post方式,当我们要传送一个较大的数据文件时,需要用post。当传递的值只需用参数方式(这个值不大于2KB)的时候,用get方式即可。

8.ajax请求时，如何解析json数据 ?

答：1.eval JSON.parse

2.JSON.parse

推荐使用第二种

9.写一个非行间样式的函数

答：function getStyle(obj,attr){

If(obj.currentStyle){

Return obj.currentStyle[attr];

} else {

Return getComputedStyle(obj,null)[attr];

}

}

10.事件委托是什么？

答：利用事件冒泡或捕获的原理，让某一子元素身上的事件由他的父元素代为执行。

11.闭包是什么，有什么特性，对页面有什么影响？

答：闭包就是能够读取其他函数内部变量的函数。由于在javascript中，只有函数内部的子函数才能读取局部变量，所以闭包可以理解成“定义在一个函数内部的函数“。在本质上，闭包是将函数内部和函数外部连接起来的桥梁。

function a(){  
           var i=0;  
           function b(){  
              alert(++i);  
           }  
              return b;  
       }

 var c = a();  
        c();

闭包的特性：

①.封闭性：外界无法访问闭包内部的数据，如果在闭包内声明变量，外界是无法访问的，除非闭包主动向外界提供访问接口；  
②.持久性：一般的函数，调用完毕之后，系统自动注销函数，而对于闭包来说，在外部函数被调用之后，闭包结构依然保存在  
系统中，闭包中的数据依然存在，从而实现对数据的持久使用。

优点：

① 减少全局变量。

② 减少传递函数的参数量

③ 封装；

闭包的缺点是过多的占用内存资源，过多的使用闭包会导致内存的溢出等；

12.添加、移除、移动、复制、创建和查找节点的方法？

答：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() 添加

removeChild() 删除

replaceChild() 替换

insertBefore(newnode,exsitnode) 插入

3.查找

getElementByTagName() 通过标签查找

getElementByName() 通过元素Name属性查找，适用于表单

getElementById() 通过id查找

getElementByClassName() 通过class名查找

4.移动

酷派项目的轮播图中，其下方的按钮有一个读条效果，每轮到那张图片，相应的按钮就会进行读条，实现该效果就是利用了，只创建一个div元素，当轮到那个图片时则把该div插入到相应的按钮中，即实现了移动。

13.jsonp为什么不是真正的ajax？

答：jsonp动态创建script标签，在回调函数中包裹数据，没有通过XMLHttpRequest对象

Ajax是页面无刷新请求数据操作

14.window load和document ready的区别？

答：window load是当页面中所有的资源全部加载完毕后执行

document ready是当Dom结构加载完成后执行

15.“==”和“===”的不同？

答：“==”只比较两个元素的值是否相等，不比较两个元素的类型，即有隐式转换的能力

“===“只有两个元素的值和数据类型都相同是才相等，否则不等，没有隐式转换

16.排序算法

答：快速排序function oSort(arr){

if(arr.length<=1){

return arr }

var left=[];

var right=[];

var oNum = Math.floor(arr.length/2);

var oNumVode = arr.splice(oNum,1);

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

if(arr[i]<oNumVode) {

left.push(arr[i]) }

else { right.push(arr[i]) } }

return oSort(left).concat([oNumVode],oSort(right))}

冒泡排序 var array = [5, 4, 3, 2, 1];

var temp = 0;

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

for (var j = 0; j < array.length - i; j++){

if (array[j] > array[j + 1]){

temp = array[j + 1];

array[j + 1] = array[j];

array[j] = temp; } } }

17.在javascript中什么是伪数组？如何将伪数组转化为标准的数组？

答：伪数组（类数组）：无法直接调用数组方法或期望length属性有什么特殊的行为，但仍可以对真正数组遍历方法来遍历它们。典型的是函数的 argument参数，还有像调用getElementsByTagName,document.childNodes之类的,它们都返回 NodeList对象都属于伪数组。可以使用Array.prototype.slice.call(fakeArray)将数组转化为真正的Array 对象。

18.javascript同源策略？

答：我们都知道JavaScript可以操作web文档的内容，试想，如果不对这一点加以限制，那么JS可以做的事情就太多了，危险性也太高，所以就针对它可以操作哪些文档的内容有了一个限制，这个限制就是同源策略。

同源策略在什么情况下会起作用呢？当web页面使用多个<iframe>元素或者打开其他浏览器窗口的时候，这一策略就会起作用。

同源策略的含义：脚本只能读取和所属文档来源相同的窗口和文档的属性。

这里就涉及到了一个浏览器如何判断两者是否同源已经如何判断脚本来源的问题。

注意一点：脚本本身的来源并不作为判断是否同源的依据，而是将脚本所属文档的来源作为判断依据。

    1. 判断脚本来源

        例如：文档A中通过script的src引用了一个外部脚本，这个脚本时google提供的，也是从google的主机上加载到文档A中的，那么这个脚本的所属文档是谁呢，答案是文档A。

    2. 判断是否同源

        理解了脚本来源，接着理解怎么判断是否同源：如果两个文档在协议、主机以及载入文档的URL端口这三点中有一点不同，就认为他们不同源。

同源策略固然在安全性上有了很大提高，但这种做法属于“宁可错杀三千，不可放过一个”。如果一个多域名站点需要在不同的子域之间共享属性，同源策略就会变得很烦人了。以下介绍三种实现“不严格的同源策略”的方法：

    1. 使用Document对象的domain属性

        默认情况下，属性domain存放的是载入文档的服务器的主机名。这一属性是可写的。

        如果两个窗口包含的脚本把domain设置为了相同的值，那么这两个窗口就不再受同源策略的约束，它们就可以相互读取对方的属性。

    2. 跨域资源共享（Cross-Origin Resource Sharing）

        这种方法用新的“Origin：”请求头和新的响应头“Access-Control-Allow-Origin”来扩展HTTP。

    3. 跨文档消息（Cross-Document Messaging）

        允许来自一个文档的脚本可以传递文本消息到另一个文档里的脚本，不管来源是否相同。

        调用window对象的postMessage()方法，可以异步传递消息事件到窗口的文档里。这种方法仅仅是一种消息传递技术。

19.在Javascript中如何检测客户端的浏览器和操作系统类型？

答：var resolution = "分辨率:" + window.screen.width + "\*" + window.screen.height;,

var ua = window.navigator.userAg

20.cookie在客户端上是如何存储的？

答：cookie是服务器暂时存放在用户电脑上的文本文件，用来让服务器辨别用户的身份；当用户在浏览网站时，web服务器就会先发送一个小资料到用户的电脑上，cookies会帮助用户在网站上所打的文字或一些选择都记录下来，当下次在访问同一个网站时，web服务器会先查看有没有它留下的cookies资料，有的话，就会依照cookie里的内容来判断使用，送出特定的网页内容给用户。

21.prompt box是什么？它的返回值有什么作用？

答：prompt() 方法用于显示可提示用户进行输入的对话框。如果用户单击提示框的取消按钮，则返回 null。如果用户单击确认按钮，则返回输入字段当前显示的文本。

在用户点击确定按钮或取消按钮把对话框关闭之前，它将阻止用户对浏览器的所有输入。在调用 prompt() 时，将暂停对 JavaScript 代码的执行，在用户作出响应之前，不会执行下一条语句。

22.如何获取JavaScript数组中最大值和最小值？

答：console.log（Math.max(1,2,3,4,5,6)）最大值

console.log（Math.min(1,2,3,4,5,6)）最小值

23.判断字符串以字母开头，后面可以是数字，下划线，字幕，长度为6-30？

答：var reg = /^[a-zA-Z\w{5,29}]$/

24.请举例使用callee属性实现函数的递归使用？

答：(function(num){

if(num <= 1) return 1;

return num\*arguments.callee(num-1);

})(5);

Callee代表了函数本身的引用

25.ajax同步和异步的区别，如何解决跨域问题？

答：同步即按步骤顺序执行，异步即同时执行互不影响

Jsonp，可以动态执行json数据处理的方法，使用js在浏览器页面上动态生成一个script标签，并加载js文件来执行

需要服务器支持

26.请解读一下Javascript代码，并指出问题所在

var Obj=function(msg){

this.msg=msg;

this.shout=function(){

alert(this.msg);

}

this.waitAndShout=function(){

setTimeout(this.shout, 2000);

}

}

var aa=new Obj("abc");

aa.waitAndShout();

答：定时器函数的执行，都是处在全局的环境中，2秒后执行的shout函数，在获取this.msg时已经取不到内容了，因为此刻的this已经代表了window对象。

var Obj=function(msg){

this.msg=msg;

var that = this;

this.shout=function(){

alert(this.msg);

}

this.waitAndShout=function(){

setTimeout(function(){

that.shout();

}, 2000);

}

}

var aa=new Obj("abc");

27.apply和call方法有什么相似之处，请说出二者的区别？

答：apply固定是两个参数，第一个是调用的对象，第二个是一个数组；

Call方法接受参数的数量不固定，第一个是对象，第二个参数数量不一定

28.给Array本地对象增加一个原型方法，他的用途是删除数组中重复的条目，并将新的数组返回？

答：Array.prototype.distinct = function(){

var arr = new Array();

var obj = {};

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

var val = this[i];

if(!obj[val]){

obj[val] = 1;

arr.push(val);

}

}

}

29.javascript是一门什么样的语言，它有那些特点？

答：我认为Jav搜索aScript 是一种[脚本语言](https://www.baidu.com/s?wd=%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-uHbvm1w9uWDdPH7bnHfz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjmvrHTkn1Ds)，官方名称为 ECMAScript（因定义语言的标准为 ECMA-262）。JS 的主要特点：1. 语法类似于常见的高级语言，如 C 和 Java；2. [脚本语言](https://www.baidu.com/s?wd=%E8%84%9A%E6%9C%AC%E8%AF%AD%E8%A8%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-uHbvm1w9uWDdPH7bnHfz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjmvrHTkn1Ds)，不需要编译就可以由解释器直接运行；3. 变量松散定义，属于[弱类型语言](https://www.baidu.com/s?wd=%E5%BC%B1%E7%B1%BB%E5%9E%8B%E8%AF%AD%E8%A8%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-uHbvm1w9uWDdPH7bnHfz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjmvrHTkn1Ds)；4. [面向对象](https://www.baidu.com/s?wd=%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-uHbvm1w9uWDdPH7bnHfz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjmvrHTkn1Ds)的。  
JS 最初是为网页设计而开发的，现在也是 Web 开发的重要语言。它支持对浏览器（浏览器对象模型，BOM）和 HTML 文档（[文档对象模型](https://www.baidu.com/s?wd=%E6%96%87%E6%A1%A3%E5%AF%B9%E8%B1%A1%E6%A8%A1%E5%9E%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-uHbvm1w9uWDdPH7bnHfz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjmvrHTkn1Ds)，DOM）进行操作，而使网页呈现动态的交互特性。  
严格的说，JS 只是 ECMAScript 的一种实现，是 ECMAScript 和 BOM、DOM 组成的一种 Web 开发技术。

Javascript的特点：  
一、简单性：  
JavaScript是一种基于Java基本语句和控制流之上的简单而紧凑的设计，从而对于学习Java是一种非常好的过渡。它的变量类型是采用弱类型，并未使用严格的数据类型。  
  
二、动态性：  
avaScript是动态的，它可以直接对用户或客户输入做出响应，无须经过Web服务程序。它对用户的反映响应，是采用以事件驱动的方式进行的。所谓事件驱动，就是指在主页(Home Page)中执行了某种操作所产生的动作，就称为“事件”(Event)。比如按下鼠标、移动窗口、选择菜单等都可以视为事件。当事件发生后，可能会引起相应的事件响应。  
三、跨平台性：  
JavaScript是依赖于浏览器本身，与操作环境无关，只要能运行浏览器的计算机，并支持JavaScript的浏览器就可正确执行。从而实现了“编写一次,走遍天下”的梦想。  
  
四、安全性：  
  
JavaScript是一种安全性语言，它不允许访问本地的硬盘，并不能将数据存入到服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互。从而有效地防止数据的丢失。

30.希望获得页面中所有的checkbox怎么做？

答：var arr = new Array();

var boxes = document.getElementsByTagName(“input”);

for(var index in boxes){

if(boxes[index].type == ‘checkbox’){

arr.push(boxes[index];

}

}

Return arr;

31.当一个Dom节点被点击时候，我们希望能够执行一个函数，应该怎么做？

答：box.onclick = function(){}

box.addEventListener("click",function(){},false);

function xxx(){}

<button onclick="xxx()"></button>

32.看下列代码输出为何？解释原因？

1.var a;

2.alert(typeof a);

3.alert(b);

答：序号2的打印undefined 因为变量a没有赋值

序号3的直接报错，因为变量b没有声明

33.看下列代码，输出什么？解释原因？

var a = null;

alert(typeof a)

答：输出object，因为所有对象的默认值都是null；

34.看下列代码，输出什么？解释原因？

var foo = “11”+2-“1”；

console.log(foo);

console.log(typeof foo);

答：第一个输出111；第二个输出number；

35.讲字符串“<tr><td>{$id}</td><td>{$name}</td></tr>”中的{id}替换成10，{$name}替换成Tony（使用正则表达式）

答：function escapeHtml(str){

return str.replace（/[<>”&]/g,function(match){

switch(match){

case “<”: return “&lt”;

case “>”:return “&gt”;

case “&”:return “&amp”;

case “\”:return “&quot”;

}

}）

}

37.看下列代码，将会输出什么？（变量声明提升）

var foo = 1 ;

(function(){

var foo;

console.log(foo);

foo = 2 ;

console.log(foo);

})();

答：输出undefined和2；函数声明与变量声明会被JavaScript引擎隐式地提升到当前作用域的顶部，但是只提升名称不会提升赋值部分。

38.有这样一个URL ：<http://item.taobao.com/item.html?a=1&b=2&c=&d=xxx&e>请写一段JS程序提取URL中的各个GET参数（参数名字和参数个数不定），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}。

答：var url=”<http://item.taobao.com/item.html?a=1&b=2&c=&d=xxx&e>”;

var paramstr=url.substring(url.indexOf(“?”)+1);

var pms = paramstr.split(“&”);

var obj = {};

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

if(pms[i].indexOf(“=”)){

obj[pms[i]] = “undefined”;

} else {

Var pa = pms[i].split(“=”);

obj[pa[0]] = pa[1];

}

}

console.log(obj.toSource());

39.看下面代码，给出输出结果？

for(var i=1;i<=3;i++){

setTimeout(function(){

console.log(i);

},0)

}

答：4 4 4

原因：定时器属于异步操作，js线程运行时会跳过执行后面的代码。那么如何让上述代码输出1 2 3？;

for(let i=1;i<=3;i++){

setTimeout(function(a){

console.log(a);

}),(i),0);

}

40.写一个function，清除字符串前后的空格（兼容所有浏览器）

答：String.prototype.trim = function(){

Return this.replace(/^\s+/,””).replace(/\s+$/,””);

}

41.下面这个ul，如何点击每一列的时候alert其index？（闭包）

<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>

答：

*var* arrLi = document.getElementById("test").children;

for (*var* k = 0; k < arrLi.length; k++) {

arrLi[k].onclick = (*function* (*num*) {

return *function* () {

*console*.log(num);

};

})(k);

}

42.给String对象定义一个repeatify方法。该方法接收一个整数参数，作为字符串重复的次数，最后返回重复制定次数的字符串，例如：

console.log(“hello”,repeatify(3));

输出为：hellohellohello；

*String*.prototype.repeatify = *String*.prototype.repeatify || *function* (*times*) {

*var* str = '';

for (*var* i = 0; i < times; i++) {

str += this;

}

return str;

};

*console*.log('hello'.repeatify(3));

这里应该考察的是对proptotype，也就是原型的理解。解释下，加深影响！！String.prototype.repeatify相当于给String的原型添加一个自定义的repeatify方法，function（time）这个函数则是方法，

首先定义个str，也就是一个空字符串，用来接受传参。repeatify(3)就是要复制3份输出，那么就遍历下吧，有几次，就加几份，这里传入参数3次，那么for循环3次，让str加上this，这里的this就是指代的自身，就是str。等于str=str+str+str，最后return，将str作为值输出，记得今天看到一句话，**函数就是找值！！！！！**这句话很重要，就是要用值做别的事情！！

还有个关键点，就是||判断！！！如果有该方法，那就跳过，使用原方法。

43.以下那条语句会产生运行错误：

A．var obj = （）； B.var obj = [] ; C.var obj = {} ; D.var obj = // ;

答：A选项会报错，没有这种写法。 B为空数组，C为空对象，D为正则

44.给Array原型添加一个方法，这个方法接收一个参数，如果这个参数在数组中存在那么就把他从数组中删除。

答：

*Array*.prototype.remove = fucntion(obj) {

for (*var* i in this) {

if (typeof this[i] == "object") {

*var* flag = true;

for (*var* attr in this[i]) {

if (this[i][attr] != obj[attr]) {

flag = false;

break;

}

}

if (flag) {

this.splice(i, 1);

}

} else {

if (this[i] == obj) {

this.splice(i, 1);

}

}

}

}

45.对下面这个对象进行克隆

var student = {name:”zhang3”,sno:”2080328”,.age:”20”,call:function(){alert(this.name + this.sno + this.age);}};

答：将student传入a函数中；

*function* a(*x*){

*var* obj = {};

for(*var* i in x){

obj[i] = x[i]

}

return obj;

}

46.请问p.name是多少？

*var* fun = *function* () {

this.name = 'peter';

return {

name: 'jack'

};

}

*var* p = new fun();

答：jack

47.请问a.info.name和b.info.name分别是多好？

*var* fun = *function* () {

}

*fun*.prototype = {

info: {

name: 'peter',

age: 25

}

}

*var* a = new fun();

*var* b = new fun();

a.info.name = 'jack';

b.info.name = 'tom';

答：都是tom；

48.写一个traverse函数，输出页面所有宽度和高度大于50像素的节点。

答：

*function* traverse(){

return *Array*.prototype.filter.call(

document.querySelectorAll('body \*'),

*function*(*node*){

return node.offsetWidth > 50 && node.offsetHeight > 50;

})

}

49.填写内容让给a.name等于‘name1’让b.name等于‘name2’。

*function* obj(*name*) {

1

}

obj. 2 = "name2";

*var* a = obj("name1");

*var* b = new obj();

答：

//创建全局函数

*function* obj(*name*){

if(name){

this.name = name;

}

return this ; //返回this引用

}

*obj*.prototype.name = "name2";//设置原型对象

*var* a = obj('name1');//直接调用函数，this等于window，a等于window

*var* b = new obj ;//等价于var b = new obj();调用函数实例化对象，this指向obj

50. javascript 语言特性中，有很多方面和我们接触的其他编程语言不太一样，比如说，javascript语言实现继承机制的核心就是  1  (原型)，而不是Java语言那样的类式继承。Javascript 解析引擎在读取一个Object的属性的值时，会沿着  2  (原型链)向上寻找，如果最终没有找到，则该属性值为  3  undefined； 如果最终找到该属性的值，则返回结果。与这个过程不同的是，当javascript解析引擎执行“给一个Object的某个属性赋值”的时候，如果当前Object存在该属性，则改写该属性的值，如果当前的Object本身并不存在该属性，则赋值该属性的值 。

51．

*var* a = *function* b() {

b = 3;

*console*.log(b);

}

a();

//function b 看做一个有名字的 匿名函数

//一个有名字的匿名函数，其函数名 是只读的，不可更改

//同时，该函数名只能在函数内部使用，相当于局部变量

//b(); 报错，无法调用，找不到b函数

52.

if (!("a" in window)) {

*var* a = 1;

}

alert(a);

//结果为undefined

53.

*var* a = 1,

b = *function* a(*x*) {

x && a(--x);

};

*console*.log(a);

答：结果为 1 ；

54.

常规模式下

*function* b(*x*, *y*, *a*) {

arguments[2] = 10;

alert(a);

}

b(1, 2, 3);

严格模式下

*function* b(*x*, *y*, *a*) {

'use strict';

arguments[2] = 10;

*console*.log(a);

}

b(1, 2, 3);

*function* fun(*a* = 55){

arguments[0] = 99;

*console*.log(a); //结果为10；

}

fun(10);

答：结果为10；严格模式下为 3；

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/arguments