**Q1：javascript的typeof返回哪些数据类型**

答案：基本数据类型：String,boolean,Number,Undefined, Null

引用数据类型：Object(Array,Date,RegExp,Function)

**------------------------------------------------------------------------------**

**Q2：如何判断某变量是否为数组数据类型？**

答案：方法一.判断其是否具有“数组性质”，如slice()方法。可自己给该变量定义slice方法，故有时会失效

方法二.obj instanceof Array 在某些IE版本中不正确

方法三.方法一二皆有漏洞，在ECMA Script5中定义了新方法Array.isArray(), 保证其兼容性，最好的方法如下：

if(typeof Array.isArray==="undefined"){

Array.isArray = function(arg){

return Object.prototype.toString.call(arg)==="[object Array]"} }

**------------------------------------------------------------------------------**

**Q3.例举3种强制类型转换和2种隐式类型转换？**

答案：强制（parseInt,parseFloat,number）

隐式（== - ===）

**------------------------------------------------------------------------------**

**Q4.split() join() 的区别**

答案：前者是切割成数组的形式，后者是将数组转换成字符串

**------------------------------------------------------------------------------**

答案：Push()尾部添加 pop()尾部删除 Unshift()头部添加 shift()头部删除

**Q5.数组方法pop() push() unshift() shift()**

**------------------------------------------------------------------------------**

**Q6：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？**

答案：直接在DOM里绑定事件：<div onclick=”test()”></div>

在JS里通过onclick绑定：xxx.onclick = test

通过事件添加进行绑定：addEventListener(xxx, ‘click’, test)

**------------------------------------------------------------------------------**

**Q7：Javascript的事件流模型都有什么？**

答案：“事件冒泡”：事件开始由最具体的元素接受，然后逐级向上传播

“事件捕捉”：事件由最不具体的节点先接收，然后逐级向下，一直到最具体的

“DOM事件流”：三个阶段：事件捕捉，目标阶段，事件冒泡

**------------------------------------------------------------------------------**

**Q8.IE和DOM事件流的区别**

答案：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

**------------------------------------------------------------------------------**

答案：Var ev = ev || window.event

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

Var target = ev.srcElement||ev.target

**Q9.IE和标准下有哪些兼容性的写法**

**------------------------------------------------------------------------------**

**Q10.ajax请求的时候get 和post方式的区别**

答案：1.一个在url后面 一个放在虚拟载体里面

2.有大小限制

3.安全问题

4.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

**------------------------------------------------------------------------------**

**Q11.call和apply的区别**

答案：Object.call(this,obj1,obj2,obj3)

Object.apply(this,arguments)

apply固定是两个参数，第一个是调用的对象，第二个是一个带有下标的数组  
call接受的参数是不固定的，第一个是对象 ，第二个参数数量不确定

**------------------------------------------------------------------------------**

答案：使用eval parse 鉴于安全性考虑 使用parse更靠谱

**Q** **12.ajax请求时，如何解释json数据**

**------------------------------------------------------------------------------**

**Q13：Ajax同步和异步的区别，如何解决跨域问题?**

答案：

　　同步：脚本会停留并等待服务器发送回复然后再继续

　　异步：脚本允许页面继续其进程并处理可能的回复

　跨域问题简单的理解就是因为JS同源策略的限制，a.com域名下的JS无法操作b.com或c.a.com下的对象，具体场景如下：

Ajax - amy - 哈哈镜

PS：(1)如果是端口或者协议造成的跨域问题前端是无能为力的

　(2) 在跨域问题上，域仅仅通过URL的首部来识别而不会尝试判断相同的IP地址对应的域或者两个域是否对应一个IP

　前端对于跨域的解决办法：

　(1) document.domain+iframe

　(2) 动态创建script标签

**------------------------------------------------------------------------------**

**Q** **14.写一个获取非行间样式的函数**

答案：function getStyle(obj,attr,value) {

if(!value) { if(obj.currentStyle) {

return obj.currentStyle(attr)} else{ obj.getComputedStyle(attr,false) }

} else{ obj.style[attr]=value} }

解释：

**------------------------------------------------------------------------------**

**Q** **15.事件委托是什么**

答案：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！

**------------------------------------------------------------------------------**

**Q16.闭包是什么，有什么特性，对页面有什么影响**

答案：闭包就是能够读取其他函数内部变量的函数。

**------------------------------------------------------------------------------**

**Q** **17.如何阻止事件冒泡和默认事件**

答案：canceBubble return false

**------------------------------------------------------------------------------**

**Q** **18.** **添加、移除、移动、复制、创建和查找节点的方法**

答案：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() //移除

replaceChild() //替换

insertBefore() //插入

3）查找

getElementsByTagName() //通过标签名称

getElementsByName() //通过元素的Name属性的值

getElementById() //通过元素Id，唯一性

**------------------------------------------------------------------------------**

**Q** **19.解释jsonp的原理，以及为什么不是真正的ajax**

答案：jsonp动态创建script标签，回调函数

Ajax是页面无刷新请求数据操作

**------------------------------------------------------------------------------**

**Q20.javascript的本地对象，内置对象和宿主对象**

答案：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

**------------------------------------------------------------------------------**

**Q21.document load 和document ready的区别**

答案：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)

**------------------------------------------------------------------------------**

**Q22.”==”和“===”的不同**

答案：前者会自动转换类型 后者不会

**------------------------------------------------------------------------------**

答案：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

**Q23.javascript的同源策略**

**------------------------------------------------------------------------------**

**Q24.编写一个数组去重的方法**

答案：function oSort(arr) { var result ={};

var newArr=[];

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

if(!result[arr[i]]) {

newArr.push(arr[i])

result[arr[i]]=1 } }

return newArr }

**------------------------------------------------------------------------------**

**Q26.排序算法**

答案：

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; } } }

**------------------------------------------------------------------------------**

**Q27：希望获取到页面中所有的checkbox怎么做？(不使用第三方框架)**

答案： var domList = document.getElementsByTagName(‘input’)

var checkBoxList = [];

var len = domList.length;　　//缓存到局部变量

while (len--) {　　//使用while的效率会比for循环更高

if (domList[len].type == ‘checkbox’) {

checkBoxList.push(domList[len]);

}

}

**------------------------------------------------------------------------------**

**Q28：什么是Ajax和JSON，它们的优缺点。**

答案： Ajax是异步JavaScript和XML，用于在Web页面中实现异步数据交互。

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）

**------------------------------------------------------------------------------**

**Q29：有这样一个URL：**

**http://item.taobao.com /item.htm?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}。**

答案： function serilizeUrl(url) {

var result = {};

url = url.split("?")[1];

var map = url.split("&");

for(var i = 0, len = map.length; i < len; i++) {

result[map[i].split("=")[0]] = map[i].split("=")[1];

}

return result;}

**------------------------------------------------------------------------------**

**Q30：正则表达式构造函数var reg=new RegExp(“xxx”)与正则表达字面量var reg=//有什么不同？匹配邮箱的正则表达式？**

答案：当使用RegExp()构造函数的时候，不仅需要转义引号（即\”表示”），并且还需要双反斜杠（即\\表示一个\）。使用正则表达字面量的效率更高。

邮箱的正则匹配：

1 var regMail = /^([a-zA-Z0-9\_-])+@([a-zA-Z0-9\_-])+((.[a-zA-Z0-9\_-]{2,3}){1,2})$/;

**------------------------------------------------------------------------------**

**Q31：.Javascript中callee和caller的作用？**

答案：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。

**------------------------------------------------------------------------------**

**Q32：** **下面这个ul，如何点击每一列的时候alert其index?（闭包）？1 <ul id=”test”>**

**2 <li>这是第一条</li>**

**3 <li>这是第二条</li>**

**4 <li>这是第三条</li>**

**5 </ul>**

答案：1 // 方法一：

var lis=document.getElementById('2223').getElementsByTagName('li');

for(var i=0;i<3;i++)

{

lis[i].index=i;

lis[i].onclick=function(){

alert(this.index);

};

}

//方法二：

var lis=document.getElementById('2223').getElementsByTagName('li');

for(var i=0;i<3;i++)

{

lis[i].index=i;

lis[i].onclick=(function(a){

return function() {

alert(a);

}

})(i);

}

**Q33：.** **给String对象添加一个方法，传入一个string类型的参数，然后将string的每个字符间加个空格返回，例如：**

**addSpace(“hello world”) // -> ‘h e l l o w o r l d’**

**------------------------------------------------------------------------------**

答案：String.prototype.spacify = function(){

return this.split('').join(' ');

};

**Q34：函数声明与函数表达式的区别？**

答案：在Javscript中，解析器在向执行环境中加载数据时，对函数声明和函数表达式并非是一视同仁的，解析器会率先读取函数声明，并使其在执行任何 代码之前可用（可以访问），至于函数表达式，则必须等到解析器执行到它所在的代码行，才会真正被解析执行。

**------------------------------------------------------------------------------**

**Q35：在Javascript中什么是伪数组？如何将伪数组转化为标准数组？**

答案：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。

**------------------------------------------------------------------------------**

**Q36：想实现一个对页面某个节点的拖曳？如何做？（使用原生JS）**

答案：1. 给需要拖拽的节点绑定mousedown, mousemove, mouseup事件

2. mousedown事件触发后，开始拖拽

3. mousemove时，需要通过event.clientX和clientY获取拖拽位置，并实时更新位置

4. mouseup时，拖拽结束

5. 需要注意浏览器边界的情况

**------------------------------------------------------------------------------**

**Q37：var numberArray = [3,6,2,4,1,5]; （考察基础API）**

**1) 实现对该数组的倒排，输出[5,1,4,2,6,3]**

**2) 实现对该数组的降序排列，输出[6,5,4,3,2,1]**

答案： var numberArray = [3,6,2,4,1,5];

numberArray.reverse(); // 5,1,4,2,6,3

numberArray.sort(function(a,b){ //6,5,4,3,2,1

return b-a;

})

**------------------------------------------------------------------------------**

**Q38：.将字符串”<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）**

答案：”<tr><td>{$id}</td><td>{$id}\_{$name}< /td></tr>”.replace(/{\$id}/g, ’10′).replace(/{\$name}/g, ‘Tony’);

**------------------------------------------------------------------------------**

**Q39：foo = foo||bar ，这行代码是什么意思？为什么要这样写？**

答案：if(!foo) foo = bar; //如果foo存在，值不变，否则把bar的值赋给foo。

短路表达式：作为”&&”和”||”操作符的操作数表达式，这些表达式在进行求值时，只要最终的结果已经可以确定是真或假，求值过程便告终止，这称之为短路求值。

**------------------------------------------------------------------------------**

**Q40：为了保证页面输出安全，我们经常需要对一些特殊的字符进行转义，请写一个函数escapeHtml，将<, >, &, “进行转义**

答案：function escapeHtml(str) {

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

switch (match) { case “<”: return “&lt;”; case “>”: return “&gt;”; case “&”: return “&amp;”; case “\””: return “&quot;”;

}})}

**------------------------------------------------------------------------------**

**Q41：写一个function，清除字符串前后的空格。（兼容所有浏览器）**

答案：if (!String.prototype.trim) {

String.prototype.trim = function() {

return this.replace(/^\s+/, "").replace(/\s+$/,"");

} }

// test the function

var str = " \t\n test string ".trim();

alert(str == "test string"); // alerts "true"

**------------------------------------------------------------------------------**

**Q42：Javascript中callee和caller的作用？**

答案：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。

**------------------------------------------------------------------------------**

**Q43：谈一谈你对javascript的理解，javascript能做什么**

答案：javascript是主要用在web中的一种动态的解析型语言。

可以改变网页里的数据，根据用户点击等操作告诉服务器可以动态的改变网页的大小。 JavaScript是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言,是一种轻量级的编程语言。同时也是一种广泛用于客户端Web开发的脚本语言，常用来给HTML网页添加动态功能，比如响应用户的各种操作,可以将动态的文本放入 HTML 页面, 可以对事件作出响应, 可以读写 HTML 元素, 可被用来验证数据, 可被用来检测访问者的浏览器, 可被用来创建 cookies

**------------------------------------------------------------------------------**

**Q44：在javascript中如何检测客户端的浏览器和操作系统类型**

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

var ua = window.navigator.userAgent

**------------------------------------------------------------------------------**

**Q45：Cookie在客户机上是如何存储的**

答案：Cookies就是服务器暂存放在你的电脑里的文本文件，好让服务器用来辨认你的计算机。当你在浏览网站的时候，Web服务器会先送一小小资料放在你的计算机上，Cookies 会帮你在网站上所打的文字或是一些选择都记录下来。当下次你再访问同一个网站，Web服务器会先看看有没有它上次留下的Cookies资料，有的话，就会依据Cookie里的内容来判断使用者，送出特定的网页内容给你。

**------------------------------------------------------------------------------**

**Q46：Prompt box是什么？它的返回值有什么用？**

答案：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。

**------------------------------------------------------------------------------**

**Q47：如何获取javascript所能表示的最大值和最小值？**

答案：alert(Math.max.apply(null, a));//最大值

alert(Math.min.apply(null, a));//最小值

**------------------------------------------------------------------------------**

**Q48：Null和Undefined有和区别与联系**

答案：null :表示无值;

undefined : 表示一个未声明的变量，或已声明但没有赋值的变量，或一个并不存在的对象属性。

**------------------------------------------------------------------------------**

**Q49：如何理解javascript中的variable typing，试用代码说明**

答案：js的基本数据类型:number（数字类型）,string（字符串）,boolean（布尔类型 true/false） //null（空类型只有一个值null）在使用typeof测试类型时，返回object。

//undefine（未定义类型）未定义类型只有一个值underfine

**------------------------------------------------------------------------------**

**Q50：如何理解javascript中的对象**

答案：JavaScript 中的所有事物都是对象：字符串、数值、数组、函数... 用function定义

**------------------------------------------------------------------------------**