1. Ajax，支持xml、html、json、jsonp都支持
2. 下面哪些语句可以 在JS里判断一个对象oStringObject是否为String。

oStringObject instanceof String这个是对的

typeof oStringObject == 'string'题目里说对象什么什么，所以这里很狡猾是object

1. 存储技术

浏览器端：

cookie

WebStorage(localStorage、sessionStorage)

userData

indexedDB

Websql

服务器端：

session

（4）如何规避javascript多人开发函数重名问题。

根据不同的开发人员实现的功能，在函数名**加前缀**

每个开发人员都把自己的函数封装到类中，然后调用的时候即使函数名相同，但是因为是要类.函数名来调用，所以也减少了重复的可能性

（5）<dl>标记定义了一个定义列表，定义列表中的条目是通过使用<dt>标记(“definition title”，定义标题)和<dd>标记(“definition description”，定义描述)创建的。<dt>给出了术语名，<dd>标记给出了术语的定义。下面这样才是对的

<dl>

<dt> </dt>

<dd> </dd>

</dl>

（6）在bootstrap中， 关于导航条，下列说法正确的是?

应该将被包裹的元素放到navbar-collapse类中（导航倒塌）

表单应该放置于navbar-form内

可以使用navbar-left和navbar-right来对齐导航条

可以使用navbar-fixed-top和navbar-fixed-bottom来将导航条固定到顶部或底部

（7）

Firefox 不支持 DOM 对象的 outerHTML innerText outerText 属性。innerText IE支持，FIREFOX不支持，新版火狐也支持。

在 IE6 IE7 IE8 中，createElement 方法不仅可以通过合法的标签名创建节点对象，还可以通过传入一段合法的 HTML 代码字符串作为参数创建节点对象。innerText IE支持，FIREFOX支持

IE6 IE7 IE8(Q) 中无法通过 "Element.setAttribute("class", "AttributeValue")" 设置元素的 class 属性，

而需要使用 "Element.setAttribute("className", "AttributeValue")"。

FIREFOX支持，IE不支持

IE6 IE7 IE8(Q) 中无法通过诸如 "Element.setAttribute("onclick", "alert('ok')")" 为元素绑定事件；

setAttribute('class'，'styleClass') FIREFOX支持，IE不支持

（8）下面有关javascript内部对象的描述，正确的有？

History 对象包含用户（在浏览器窗口中）**访问过的 URL**

Location 对象包含有关**当前 URL 的信息**

Window 对象表示**浏览器中打开的窗口**

Navigator 对象包含有关**浏览器的信息**

（9）假设当前屏幕分别率为1024×768，定义一个居中的占屏幕一半大小的表格的语句是

<TABLE ALIGN=”CENTER” WIDTH=”50%”></TABLE>

<TABLE ALIGN=”CENTER” WIDTH=”512″></TABLE>

<DIV ALIGN=”CENTER”><TABLE WIDTH=”512″></TABLE></DIV>

<CENTER><TABLE WIDTH=”50%”></TABLE></CENTER>

（10）嵌入在HTML文档中的图像格式可以是？

\*.gif可以

\*.tif不行

\*.bmp可以

\*.jpg可以

（11）bootstrap中关于h标签

<h1>h1. Bootstrap heading <small>Secondary text</small></h1>

<h1>最大

<small>表示副标题

（12）下面属于CSS3新增属性的有？

box-shadow

text-shadow

border-radius

rgba

（13）  
请给Array本地对象增加一个原型方法，它用于删除数组条目中重复的条目(可能有多个)，返回值是一个包含被删除的重复条目的新数组。

Array.prototype.distinct = function() {

 var ret = [];

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

    for (var j = i+1; j < this.length;) {

        if (this[i] === this[j]) {

            ret.push(this.splice(j, 1)[0]);

        } else {

            j++;

        }

    }

 }

 return ret;

}

//for test

alert(['a','b','c','d','b','a','e'].distinct());

（14）

请填充代码，使mySort()能使传入的参数按照从小到大的顺序显示出来。

function mySort() {

    var tags = new Array();//使用数组作为参数存储容器

    请补充你的代码

    return tags;//返回已经排序的数组

}

var result = mySort(50,11,16,32,24,99,57,100);/传入参数个数不确定

console.info(result);//显示结果

补充的代码：

function mySort() {

var tags = new Array();

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

tags.push(arguments[i]);

}

tags.sort((a,b)=>a-b);

return tags;

}

var result = mySort(50,11,16,32,24,99,57,100);

console.info(result);

1. 用javascript实现用户登录验证的代码。

<script language=javascript>

function checkSubmit()

{

**if** ((document.form1.name.value)==”)

    {

        window.alert (‘姓名必须填写’);

        document.form1.name.select();

        document.form1.name.focus();

**return** **false**;

    }

**else**

**return** **true**;

}

</script>

<form name=”form1″ onsubmit=”javascript:**return** checkSubmit()”>

<input type=”text” name=”name”>

</form>