**1.简述div元素和span元素的区别。**

div是块元素，默认换行，一般用于排版。

span是行内元素，默认不换行，一般用于局部文字样式。

通过display属性他们之间可以相互转换，div默认为block,span默认为inline。

**2.谈谈javascript数组方法sort()的使用，重点介绍sort()参数的使用及其内部机制。**

语法：arrayObject.sort(sortby)

参数sortby:可选，规定排序顺序。必须是函数。

返回对数组的引用。注意，数组在原数组上进行排序，不生成副本。

如果该方法没有使用参数，将按照字符编码的顺序进行排序。

如果想要按照其他标准进行排序，就需要提供比较函数，该函数要比较两个值，然后返回一个用于说明这两个值的相对顺序的数字。比较函数应该具有两个参数a和b，其返回值如下：

1）若a小于b，在排序后的数组中a应该出现在b之前，返回一个小于0的值。

2）若a等于b，返回0。

3）若a大于b，则返回一个小于0的值。

例：按照数值进行排序

<script type="text/javascript">

function sortNumber(a,b)

{

return a - b

}

var arr = new Array(6)

arr[0] = "10"

arr[1] = "5"

arr[2] = "40"

arr[3] = "25"

arr[4] = "1000"

arr[5] = "1"

document.write(arr + "<br />")

document.write(arr.sort(sortNumber))

</script>

**3.结合<span id="outer"><span id="inner">text</span></span>这段结构，谈谈innerHTML outerHTML innerText之间的区别。**

innerHTML:对象里面的HTML内容，这里是<span id="inner">text</span>

outerHTML:对象里面包括对象本身的HTML内容，这里是<span id="outer"><span id="inner">text</span></span>

innerText:对象里面的文本内容，这里是text

**4.说几条XHTML规范的内容。**

1）标签名必须用小写字母。

2）所有标签都必须被关闭，包括空标签。

3）属性值必须加引号。

4）用Id属性代替name属性。

5）所有XHTML文档必须进行文件类型声明。

**5.对Web标准化（或网站重构）知道哪些相关知识，简述几条你知道的Web标准？**

为什么要使用web标准？

1）具有更好的兼容性。

2）易于代码维护和开发。

3）可增加网站的访问量。

4）标准的web文档更易被转换为其他格式。

5）更易被搜索引擎访问，也更易被准确索引。

6）更易被JavaScript和DOM代码访问。

W3C标准：HTML、XHTML、CSS、XML、XSL、DOM。

**6.完成foo()函数的内容，要求能够弹出对话框提示当前选中的是第几个单选框。**

<html>

<head>

    <script type="text/javascript">

        function foo() {

    //在此处添加代码

            var radioGroup = document.getElementsByName("radioGroup");

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

                if(radioGroup[i].checked) {

                    alert("你选择了第" + (i+1)+"单选框");

                }

            }

            return false;

        }

    </script>

</head>

<body>

<form name="form1" onsubmit="return foo();">

    <input type="radio" name="radioGroup"/>

    <input type="radio" name="radioGroup"/>

    <input type="radio" name="radioGroup"/>

    <input type="radio" name="radioGroup"/>

    <input type="radio" name="radioGroup"/>

    <input type="radio" name="radioGroup"/>

    <input type="submit"/>

</form>

</body>

</html>

**7.填充注释部分的函数体，使得foo()函数调用弹出“成功”的对话框。代码应尽量简短。**

<html>

<head>

    <script>

    function foo() {

        var str = reverse('a,b,c,d,e,f,g');

        alert(str);

        if (str == 'g,f,e,d,c,b,a') {

            alert('成功');

        }else {

            alert('失败');

        }

    }

    function reverse(str)

    {

        // 在此处加入代码，完成字符串翻转功能

        return str.split(",").reverse().join(",");

    }

    </script>

</head>

<body>

    <INPUT type="button" value="str" onclick="foo()"/>

</body>

</HTML>

8.下面代码：

<script type="text/javascript">

var x = 1;

var y = 0;

var z = 0;

function add(n) {

n = n +1;

}

y = add(x);

function add(n) {

n = n + 3;

}

z = add(x);

s = y + z;

</script>

求：y、z、s的值。

答案：undefined、undefined、NaN。两个add函数都没有返回值，所以是undefined，两个undefined相加自然是NaN。

9.下面代码：

<script type="text/javascript">

var x = 1;

var y = 0;

var z = 0;

function add(n) {

return n = n +1;

}

y = add(x);

function add(n) {

return n = n + 3;

}

z = add(x);

s = y + z;

</script>

求：y、z、s的值。

答案：4，4，8。直接通过function声明的函数，后面定义的会影响前面的。

<script type="text/javascript">

function x() {alert("2");}

x();//输出3

function x() {alert("3");}

x();//输出3

</script>

通过var声明的函数，后面定义的不会影响前面的：

<script type="text/javascript">

var x = function() {alert("0");}

x();//输出0

var x = function() {alert("1");}

x();//输出1

</script>

通过function定义的函数，后面定义的，照旧影响前面的引用，但是不能改变通过var声明函数后的引用，反而，通过var声明的函数，改变了后来通过声明函数之后的引用。

<script type="text/javascript">

var x = function() {alert("1");}

x();//输出1

function x() {alert("3");}

x();//输出1

</script>

<script type="text/javascript">

function x() {alert("2")};

x();//输出3

var x = function() {alert("0");}

x();//输出0

var x = function() {alert("1");}

x();//输出1

function x() {alert("3");}

x();//输出1

</script>

10.下面的输出结果为：

typeof(NaN) //number

typeof(Infinity) //number

typeof(null) //object

typeof(undefined) //undefined

NaN == NaN //false

NaN != NaN //true

NaN >= NaN //false

null == undefined //true

null >= undefined //false

null <=undefined //false

parseInt("123abc") //123

"123abc" - 0 //NaN

Infinity > 10 //true

Infinity == NaN //false

Infinity > "abc" //false

true == 1 //true

new String("abc") == "abc" //true

new String("abc") === "abc" //false

var a = "123abc";

alert(typeof(a++)); //number

alert(a); //NaN

var a = "123abc";

a.valueOf = function(){return parseInt(a);}

a.toString = function(){return parseInt(a);}

alert(++a); //NaN

alert(a-0); //NaN

var a = new Object();

a.toString = function(){return "123abc";}

a.valueOf = function(){return parseInt(a);}

alert(++a); //124

alert(a-0); //124

String.prototype.valueOf = function()

{

return parseFloat(this);

}

alert("123abc" > 122); //false

alert(new String("123abc") > 122); //true

var s = new String("abc");

alert(typeof(s) == typeof("abc")); //false typeof(s) == object 而typeof("abc")==string

alert(s === "abc"); //false

alert(s.toString() == s); //true