Web前端笔试题（四）

1. 将字符串 border-left-color，moz-viewport 转换成 borderLeftColor，mozViewport格式(封装函数)

function toCamelCase( str ){

var subStr = str.split("-");

var str = subStr[0];

for( i = 1; i < subStr.length; i++ ){

str += subStr[i].substr(0,1).toUpperCase()+subStr[i].substr(1);

}

return str;

}

alert( toCamelCase('border-left-color') );

1. 输出明天的日期

<script type="text/javascript">

function GetDateStr(AddDayCount) {

var dd = new Date();

dd.setDate(dd.getDate()+AddDayCount);//获取 AddDayCount天后的日期

var y = dd.getFullYear();

var m = dd.getMonth()+1;//获取当前月份的日期

var d = dd.getDate();

return y+"-"+m+"-"+d;

}

document.write("前天："+GetDateStr(-2));

document.write("<br />昨天："+GetDateStr(-1));

document.write("<br />今天："+GetDateStr(0));

document.write("<br />明天："+GetDateStr(1));

document.write("<br />后天："+GetDateStr(2));

document.write("<br />大后天："+GetDateStr(3));

</script>

1. 完成下列功能
2. 为字符串扩展删除左侧、右侧及左右两侧空格

1.消除字符串左边的空格

function leftTrim(str){

return str.replace(/^\s\*/,"");//^符号表示从开头即左边进行匹配

}

2.消除字符串右边的空格

function rightTrim(str){

return str.replace(/\s\*$/,"");

}

3.消除字符串两边的空格

function trim(str){

return str.replace(/(^\s\*)|(\s\*$)/g,"");

}

//alert("111"+trim(" aaa ")+"111");

1. 为数组扩展indexOf、remove及removeat方法

1.var oldArrayIndexOf = Array.indexOf;//判断是否原始浏览器是否存在indexOf方法

array.prototype.indexOf = function(obj) {

if(!oldArrayIndexOf) {

for(vari = 0, imax = this.length; i < imax; i++) {

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

return i;

}

}

return-1;

} else{

return oldArrayIndexOf(obj);

}

}

2. Array.prototype.remove = function (dx) {

if (isNaN(dx) || dx > this.length) {

return false;

}

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

this[i] = this[i+1];

}

this.length -= 1;

};

var arr = ['a','b','a','a'];

arr.remove( 1 );

alert(arr);

3.Array.prototype.removeAt=function(index){

this.splice(index,1);

}

1. 实现兼容的getElementsByClassName()方法

var $ = {

getEleByClass: function( cls, parent, tag ){

var parent = parent || document;

var tag = tag || "\*";

if( parent.getElementsByClassName ){

return parent.getElementsByClassName( cls );

} else {

var aClass = [];

var reg = new RegExp( "(^|\\s+ )" + cls + "(\ \s+|$)" );

var aEle = this.getEleByTag( tag, parent );

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

reg.test( aEle[i].className ) && aClass.push( aEle[i] );

}

return aClass;

}

},

getEleByTag: function( ele, obj ){

return ( obj || document ).getElementsByTagName( ele );

},

hasClass: function( ele, cls ){

return ele.className.match( new RegExp( "(^|\\s)" + cls + "(\\s|$)" ));

},

addClass: function( ele, cls ){

if( !$.hasClass( ele, cls ) ){

ele.className += " " + cls;

}

},

removeClass: function( ele, cls ){

if( $.hasClass( ele, cls ) ){

var reg = new RegExp( "(^|\\s)" + cls + "(\\s|$)" );

ele.className = ele.className.replace( reg, " " );

}

}

};

1. 实现查找对象的方法（三组）

*见题目4*

1. 实现获得和设置对象的宽和高
2. 实现给对象设置和获得属性

var a=document.getElementById("tet");

/\*

alert(a.getAttribute("title"));

a.setAttribute("name","cc");

alert(a.getAttribute("name"));\*/

1. 实现hasClass、addClass及removeClass的功能（二组）

*见题目4*

1. 根据以下xml请写出对应的json，并解析在页面上，要求以表格格式打印（要求兼容所有浏览器）（一组）

<xml>

<list>

<item>

<id>12</id>

<name>张三</name>

</item>

<item>

<id>13</id>

<name>李四</name>

</item>

</list>

</xml>

demo.json

[

{

"id": 12,

"name": "张三"

},

{

"id": 13,

"name": "李四"

}

]

var str = "";

$.ajax({

url:"demo.json",

dataType: "json",

async: false,

success:function(data){

$.each( data,function(index,value){

str+="<tr><td>"+value.cid+"</td><td>"+value.oname+"</td></tr>"

});

}

});

$('table').html(str);

1. 实现isArray、inArray功能（一组）

function isArray(obj) {

return Object.prototype.toString.call(obj) === '[object Array]';

}

//needle 待检测字符串

//haystack 数组或者是以|分割的字符串

function in\_array(needle,haystack) {

haystack=isArray(haystack)?haystack:haystack.split("|");

if(typeofneedle =='string'||typeofneedle =='number') {

for(var i in haystack) {

if(haystack[i] == needle) {

Return true;

}

}

}

return false;

}