2017.6.24

一:

Array对象

<script type="text/javascript">

function test1(){

//指定元素个数

var a1=new Array(5);

//不指定元素个数

var a2=new Array();

//动态初始值创建

var a3=new Array("AA","BB","CC","DD");

//直接量语法

var a4=["张三","李四","王五","赵六"];

}

function test2(){

var allStr=["AA","BB","CC"];

//利用动态自增长

allStr[3]="DD";

allStr[4]="EE";

alert("追加:"+allStr);

//利用concat()

allStr= allStr.concat("张三","李四");

alert("追加:"+allStr);

}

function test3(){

var allStr=["AA","BB","CC","DD","EE","FF","GG"];

//删除第一个元素,并返回删除的元素

var delStr=allStr.shift();

alert("删除后: "+allStr+" 删除的元素:"+delStr);

//删除指定范围的元素

//"BB","CC","DD","EE","FF","GG"

//allStr.splice(2,3);

allStr.splice(2,3,"zhangsan","lisi");

alert("删除后: "+allStr);

}

function test4(){

//反转: 将原有数据,按照反向排序方式 进行重新排序. 比如 : 1,2,3 反转: 3,2,1

var allStr=["AA","BB","CC","DD","EE","FF","GG"];

alert("反转之前:"+allStr);

allStr.reverse();

alert("反转之后:"+allStr);

}

function test5(){

/\* 排序: 将数组的元素按照某个标准 进行 排序

\* sort()

\*/

var allStr=["EE","AA","DD","BB","CC"];

//var allStr=["李四","张三","王五","赵六","田七"];

alert("默认元素顺序:"+allStr);

allStr.sort(myCompare);

alert("默认排序标准:"+allStr);

}

//编写一个比较器: 规范比较标准 1[后一个在前] 0[等于] -1[前一个在前]

function myCompare(e1,e2){

// 后面的字母 在前面

var e1Code=e1.charCodeAt(0);

var e2Code=e2.charCodeAt(0);

if(e1Code>e2Code){//66>65

return -1;

}else if(e1Code<e2Code){//66>65

return 1;

}else{

return 0;

}

}

function test6(){

var allStr=["AA","BB","CC","DD","EE","FF","GG"];

var newStr=allStr.slice(2,6);

alert("allStr: "+allStr);

alert("newStr: "+newStr);

}

function test7(){

//toString split(",")

var allStr=["AA","BB","CC","DD","EE"];

var str1=allStr.toString();

alert("str1: "+str1+" 字符个数:"+str1.length);

//join()

var str2=allStr.join("-");

alert("str2: "+str2+" 字符个数:"+str2.length);

}

</script>

</head>

<body>

<!--

作者: 漂流哥

日期: 2017-6-24

Array对象表示数组对象.

-->

Array对象测试 <br>

<!-- 创建数组 -->

<input type="button" value="确定1" onclick="test1()">

<!-- 添加操作 -->

<input type="button" value="确定2" onclick="test2()">

<!-- 删除操作 -->

<input type="button" value="确定3" onclick="test3()">

<!-- 反转数组 -->

<input type="button" value="确定4" onclick="test4()">

<!-- 数组排序-->

<input type="button" value="确定5" onclick="test5()">

<!-- 截取数组 -->

<input type="button" value="确定6" onclick="test6()">

<!-- 转为String -->

<input type="button" value="确定7" onclick="test7()">

</body>

</html>

二:

正则表达式的网站:

http://tools.jb51.net/regex/create\_reg

<http://tool.oschina.net/regex#>

RegExp对象

<script type="text/javascript">

function test1(){

//构造方法创建正则对象: 每个正则对象都代表一个 要求,因此在创建正则对象必须设置 正则要求.

//字符串只含有 数字: \d [0-9]

/\*

\* 只能为 6-10 位数字组成: ^[0-9]{6,10}$

\* 只能由数字和字母组成,5-10: ^[0-9a-zA-Z]{5,10}$

\* 字母开头,只能由数字和字母组成,5-10: ^[a-zA-Z][0-9a-zA-Z]{4,9}$

\*/

//var patt1=new RegExp("[0-9]");

var patt1=/^[a-zA-Z][0-9a-zA-Z]{4,9}$/

var str="a123456b";

var rs=patt1.test(str);

document.write(rs);

}

function test2(){

var info="你大爷的 替我问候你大爷的";

//检索是否含有不合法文字

var patt=/[你][大][爷]/g;

//若含有 替换不合法文字

info=info.replace(patt,"\*\_\*");

document.write(info);

}

</script>

</head>

<body>

<!--

作者: 漂流哥

日期: 2017-6-24

JS中提供RegExp类作为正则对象,处理正则要求.

-->

JS程序 <br>

<!-- 创建正则对象和基本使用 -->

<input type="button" value="确定1" onclick="test1()">

<!-- 替换不合法文字 -->

<input type="button" value="确定2" onclick="test2()">

</body>

</html>

三:

Global对象

<script type="text/javascript">

var i=1;//全局变量,实质为全局对象的和属性

function test1(){

var data1="23";

var data2=23;

var data3="张三";

var rs1=isNaN(data3);

document.write("rs1:"+rs1+"<br>");

//延伸

var newData3=Number(data3);

document.write("newData3!=0: "+(newData3!=0));

}

function test2(){

var data1=23.33333;

var data2=10/3;

var data3="zhangsan";

var rs1=isFinite(data3);

document.write("rs1:"+rs1+"<br>");

}

function test3(){

var data1="23.25";

//var newData1=Number(data1);

//var newData1=parseInt(data1);

var newData1=parseFloat(data1);

document.write(newData1+" "+typeof(newData1)+"<br>");

var data2=56;

var newData2=String(data2);

document.write(newData2+" "+typeof(newData2)+"<br>");

}

</script>

</head>

<body>

<!--

作者: 漂流哥

日期: 2017-6-24

Global对象表示全局对象,当浏览器解析JS程序时自动创建一个全局对象.

所有全局代码[全局变量 方法]都属于全局对象的代码[属性,功能]

isNaN(数据): 用于检测数据能否为数字或自动转为数字

1.若满足条件,返回false. 否则返回true

2.NaN表示转换失败,NaN不等于任何数据

isFinite(数据): 检测是否为有取值范围的数字

1.有取值范围

2.数字类型

类型转换方法: [string number boolean]

Number(): 将数据尝试转为数字. 若转换失败,为NaN

String(): 将数据尝试转为字符串.

parseInt(): 将数据尝试转为整数. 若转换失败,为NaN

parseFloat(): 将数据尝试转为小数. 若转换失败,为NaN

附加:

Boolean()

-->

JS程序 <br>

<!-- 检测是否为数字或转为数字 -->

<input type="button" value="确定1" onclick="test1()">

<!-- 检测是否为有限数字 -->

<input type="button" value="确定2" onclick="test2()">

<!-- 类型转换 -->

<input type="button" value="确定3" onclick="test3()">

</body>

</html>

数据为0或者””时为false,其余皆为true.(转为boolean)