2017.6.21

一:

|----内置对象:

JS含有很多内置对象,有些内置对象受大部分浏览器支持,有些内置对象不受浏览器支持.

Array,Boolean,Date,Error,EvalError,Function,Math,Number,Object,RangeError,ReferenceError,RegExp,String,SyntaxError,TypeError,URIError

常见内置对象:

Array,Boolean,Date,Math,Number,String,RegExp,Global,Events.

|----Math对象:

Math对象即算数对象,用于执行常见的算数任务.

常用属性:

|  |  |
| --- | --- |
| 属性名 | 作用 |
| PI | 返回圆周率 |
| E | 返回算术常量e,即自然对数的底数 |

常用方法:



|----Date对象:[用法和功能,与Java的Date对象基本相同]

Date对象用于处理日期和时间.

创建Date对象:

var myDate=new Date();  
常用方法:



|----String对象:[用法和功能,与Java的String对象基本相同]

String对象表示字符串对象,用于处理文本/字符串.

创建String对象:

1. "字符串值"
2. new String("字符串值");

常用属性:

字符串.length; 获得字符串的字符个数

常用方法:



|----Array对象:

Array对象表示数组对象.

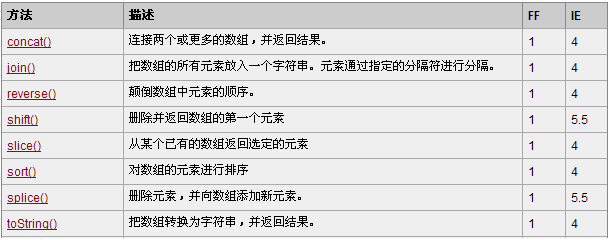
创建Array对象:

省略

常用属性:

数组.length: 获得当前数组的元素个数

常用方法:



|----RegExp对象

RegExp对象表示正则表达式,是对字符串执行模式匹配的强大工具.

创建RegExp对象:

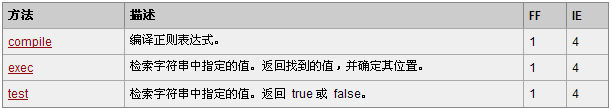
第一种: /pattern/attributes 比如: var patt=/^[a-zA-Z]{20}$/ 匹配只含有20个字母

第二种:new RegExp(pattern, attributes); 比如:var patt=new RegExp("^[a-zA-Z]{20}$") 双引号取代"/"

pattern表示模式: 模式可以是普通字符,也可以是正则表达式(不能有匹配范围,即不要第二个参数)

attributes表示匹配范围: i:大小写不敏感的匹配; g:查找所有匹配而非在找到第一个匹配后停止; m:多行匹配

常用方法:



|----Global对象:

Global对象表示JS全局对象,无法手动创建该对象.在Scripting引擎被初始化时自动创建Global对象,并立即使其方法和属性可用,即创建整体的执行环境.

常用方法:



|----Boolean对象:

Boolean对象表示布尔对象.其中,boolean表示基本数据类型,Boolean可以看着是boolean的封装类.

二:

转移字符

<script type="text/javascript">

function test1(){

for(var i=0;i<10;i++){

//代码过多,不方便维护

//降低代码的有效范围: 只在if体中有效

/\*if(i!=4&&i!=7){

document.write("i="+i+"<br>");

}\*/

if(i==4||i==7){

//结束本次循环体的执行

continue;

}

document.write("i="+i+"<br>");

}

}

function test2(){

document.write("break测试开始<br>");

/\*for(var i=0;i<10;i++){

if(i==4){

break;

}

document.write("i="+i+"<br>");

}\*/

var i=3;

switch(i){

case 1: document.write("i=1<br>"); break;

case 2: document.write("i=2<br>"); break;

case 3: document.write("i=3<br>"); break;

case 4: document.write("i=4<br>"); break;

default: document.write("i=??<br>");

}

document.write("break测试结束<br>");

/\*

\* 问题: switch case中break是必须含有的吗? 如何理解case?

\*

\*/

}

function temp(){

document.write("temp测试开始<br>");

for(var i=0;i<10;i++){

if(i==4){

return;

}

document.write("i="+i+"<br>");

}

document.write("temp测试结束<br>");

}

function test3(){

document.write("test3()测试开始<br>");

temp();

document.write("test3()测试结束<br>");

}

</script>

</head>

<body>

<!--

作者: 漂流哥

日期: 2017-6-21

JS属于基于对象的脚本语言,含有复杂的语法: 变量 数组 控制语句 循环语句等等.

JS提供了控制循环语句的关键字---转移字符: continue , break , return

continue:

1.continue中文意思: 继续 接着

2.continue可以用于JS的循环语句中,结束本次循环体的执行,直接进入下次循环的执行.

格式: continue;

break:

1.break中文意思: 打断 终止

2.break可以用于JS的循环语句和switch代码中,结束整个循环或switch代码块的执行.

格式: break;

return:

1.return中文意思: 返回 终止 结束

2.return可以用于JS的方法体中,结束整个方法的执行,并根据需要返回数据

格式: return;

-->

转移字符测试<br>

<!-- continue测试 -->

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

<!-- break测试 -->

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

<!-- return测试 -->

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

</body>

</html>

三:

Math对象

<script type="text/javascript">

function test1(){

//new Math(); Math没有构造方法[创建对象的方法],不能创建对象.

var e=Math.E;//获得自然对数的底数

alert("E: "+e);

var PI=Math.PI;//获得圆周率

alert("PI: "+PI);

}

function test2(){

document.write("abs(-56): "+Math.abs(-56)+"<br>");

document.write("pow(6,3): "+Math.pow(6,3)+"<br>");

document.write("sqrt(81): "+Math.sqrt(81)+"<br>");

document.write("sqrt(81): "+Math.sqrt(81)+"<br>");

document.write("max(23,56): "+Math.max(23,56)+"<br>");

document.write("min(23,56): "+Math.min(23,56)+"<br>");

document.write("max(23,56,88,77,100,65,123,42,11): "+Math.max(23,56,88,77,100,65,123,42,11)+"<br>");

document.write("ceil(4.36): "+Math.ceil(4.36)+"<br>");

document.write("floor(4.96): "+Math.floor(4.96)+"<br>");

document.write("round(4.96): "+Math.round(4.96)+"<br>");

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

//生成1-100之间的随机整数

document.write("random(): "+Math.round(Math.random()\*100)+"<br>");

}

function test3(){

var checkCode="";//默认随机码

for(var i=0;i<4;i++){

var num=Math.round(Math.random()\*10);

checkCode+=num;//""+3 "3"+"9"

}

alert("验证码:"+checkCode);

}

</script>

</head>

<body>

<!--

作者: 漂流哥

日期: 2017-6-21

面向对象的步骤:

1.分析需求中涉及哪些对象

2.抽象现实对象到程序中

3.通过系统模拟现实对象: new 设计图纸名();//设计图纸名==方法名==类名==对象名

4.操作模拟的对象实现需求

如何理解内置对象?

JS本身预定义的JS对象,称为内置对象.

如何使用内置对象?

模拟对象,操作对象.

-->

Math对象测试 <br>

<!-- 常用属性 -->

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

<!-- 常用方法 -->

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

<!-- 随机验证码 -->

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

</body>

</html>

四:

Date对象

<script type="text/javascript">

function test1(){

var nowDate=new Date();//自动将当前日期和时间作为 初始值

alert("当前日期:"+nowDate);

alert("本地日期:"+nowDate.toLocaleString());

alert("本地日期:"+nowDate.toLocaleDateString());

alert("本地时间:"+nowDate.toLocaleTimeString());

}

function test2(){

//2017年6月21日 星期三 17点09分38秒

var date=new Date();

var year=date.getFullYear();//获得当前年份

var month=date.getMonth()+1;//获得当前月份

var day=date.getDate();//获得号数

var weekday=date.getDay();//获得星期数

var week=["日","一","二","三","四","五","六"];

weekday=week[weekday];

var hour=date.getHours();//获得小数

var min=date.getMinutes();//获得分钟

var sec=date.getSeconds();//获得秒数

var time=year+"年"+month+"月"+day+"日 星期"+weekday+" "+hour+"点"+min+"分"+sec+"秒";

document.getElementById("time").innerHTML=time;

}

function autoTime(){

//创建一个定时器,每隔500毫秒执行1次指定代码

setInterval("test2()",500);

}

</script>

</head>

<body>

<!--

作者: 漂流哥

日期: 2017-6-21

-->

Date对象测试 <br>

<!-- 创建Date对象 和 本地时间 -->

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

<!-- Date常用方法 -->

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

<!-- 创建定时器 -->

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

<h1 id="time"></h1>

</body>

</html>