JS内置对象

- Array对象
- String对象
- Math对象
- Date对象
- RegExp对象

1. Array对象

• Array对象指的是可以存储多个相同或不同类型的值。使用Array对象存储数据之前必须先创建Array对象。

• 创建数组对象:

- 先声明后赋值:

格式1:数组对象名称 = new Array(数组元素个数)

格式2:数组对象名称 = new Array()

例如:

-声明的同时赋值:

数组对象名称 = new Array(元素1,元素2,...)

例如: fruit=neArray("apple","pear","orange");

• 数组元素的引用:

- 使用数组名可以获取整个数组的值
- -若要获取数组元素的值,则需要使用数组名,同时借助下标。数组下标从0开始,到数组长度-1结束,即第一个元素的下标为0,最后一个元素的下标为数组长度-1

例如:

fruit=new Array(3)的元素分别为: fruit[0]、fruit[1]、fruit[2]

- Array对象的常用属性:
 - -length: 获取数组长度(数组元素个数)
- Array对象的常用方法:

方法	描述
concat(数组1,, 数组n)	用于将一个或多个数组合并到数组对象中。参数可以是具体的值,也可以是数组对象。
join(分隔符)	将数组内各个元素以分隔符连接成一个字符串。参数可以是省略,省略参数时,分隔符默认为"逗号"。
push(元素1,元 素n)	向数组的末尾添加一个或多个元素,并返回新的长度。注: 必须至少有一个参数
reverse()	颠倒数组中元素的顺序。
slice(start,end)	返回包含从数组对象中截取第start~end-1之间的元素的数组。注:end参数可以省略。省略时表示从start位置开始一直到最后的元素,全部截取。
sort()	按字典顺序对数组元素重新排序。
toString()	把数组转换为字符串,并返回转换后的字符串。转换效果等效于不带参数的join()。

• 数组对象属性和方法的引用:

数组对象。属性 数组对象。方法(参数1,参数2,···)

Array对象示例

</body> </html>

```
Array数组对象的使用
                                                  fruit数组的元素个数是: 3
                                                  直接输出fruit数组的结果: 苹果.橙子.梨子
                                                  对fruit数组调用toString后的输出结果: 苹果,橙子,梨子
                                                  对fruit数组使用默认字符分隔数组的输出结果: 苹果.橙子.梨子
<!doctype html>
                                                  对fruit数使用'、'分隔数组的输出结果:苹果、橙子、梨子将两个数组连接在一起后的输出结果:苹果,橙子,梨
<html>
                                                  子,pear,apple,orange
<head>
                                                  fruit1数组排序后的输出结果: apple,orange,pear
                                                  fruit数组添加元素后的输出结果: 苹果,橙子,梨子,香焦
<meta charset="utf-8">
                                                  对fruit数组截取1~3位置的元素后的输出结果:橙子,梨子
<title>Array数组对象的使用</title>
                                                  fruit数组倒排数组元素后的输出结果: 香焦,梨子,橙子,苹果
<script type="text/javascript">
  var fruit=new Array("苹果","橙子","梨子");
                                          //创建数组
  var fruit1=new Array("pear", "apple", "orange"); //创建数组
  document.write("fruit数组的元素个数是: ",fruit.length); //访问数组对象的属性
  document.write("直接输出fruit数组的结果: ",fruit); //直接访问数组对象
  //以下代码分别演示了数组对象的各个方法的使用
  document.write("对fruit数组调用toString后的输出结果: ",fruit.toString());
  document.write("对fruit数组使用默认字符分隔数组的输出结果: ",fruit.join());
  document.write("对fruit数使用'、'分隔数组的输出结果: ", fruit.join("、"));
  document.write("将两个数组连接在一起后的输出结果: ",fruit.concat(fruit1));
  document.write("fruit1数组排序后的输出结果: ",fruit1.sort());
  fruit.push("香焦");
  document.write("fruit数组添加元素后的输出结果: ",fruit);
  document.write("对fruit数组截取1~3位置的元素后的输出结果: ",fruit.slice(1,3));
  document.write("fruit数组倒排数组元素后的输出结果: ", fruit.reverse());
</script>
<body>
```

2. String 对象

- String 对象是包装对象,用来保存和处理字符 串常数
- 建立字符串对象语法:

字符串对象名称=new String(字符串常量)

- String对象常用属性:
 - -length:用于判断字符串的字符长度
- String对象(String变素引值从0开始

方 法	描 述		
charAt(位置)	获取String对象在指定位置处的字符		
indexOf(要查找的字符串)。	获取要查找的字符串在String对象中 <mark>首次</mark> 出现的位置		
lastIndexOf(要查找的字符串)	获取要查找的字串在String对象中的最后一次出现的位置		
substr(索引值 ¹ [,长度])	从 String 对象的索引值处开始截取 String 对象的所有字串 或截取指定长度的字串		
substring(索引值I [,索引值J])	截取由索引值i到索引值j-1之间的字符串		
split(分隔符)	把 String对象中的字符串按分隔符拆分成字符串数组		
replace(需替代的字串,新字串)	用新字串替换需替代的字串		
toLowerCase()	把 String对象中的字串转换成小写字母		
toUpperCase()	把 String对象中的字串转换成大写字母		
toString()	获取String对象的字符串值		
valueOf()	获取String对象的原始值		
concat(字串1,字串2,···)	将参数中的各字符串与 String 对象中的字符串结合成一个 字符串		

• String对象属性和方法的使用:

String对象.属性 String对象.方法(参数1,参数2,···)

字符串对象的比较与字符串变量的比较

- 字符串变量的比较: 直接将两个字符串变量的值进行比较
- 字符串对象的比较:必须先使用toString()或 valueOf()方法获取字符串对象的值,然后用值进 行比较
- 例如:

```
var str1="JavaScript";
var str2="JavaScript";
var strObj1=new String(str1);
var strObj2=new String(str2);
if(str1==str2)
if(strObj1.valueOf()==strObj2.valueOf())
```

String对象示例

第一个's'的位置是: 3

```
最后一个 's'的位置是:19
<script>
                                                     字符串对象的长度是:21
                                                    数组arr中的各个元素为:
var str = new String("This is a test phrase");
                                                     This
var firstIndex= str.indexOf("s");
                                                     is
var lastIndex= str.lastIndexOf("s");
/* 以空格为分隔符,把字符串划分为字符数组*/
                                                     test
var arr= str.split(" ");
                                                    phrase
var len = str.length;
【var result = "第一个's'的位置是: "+ firstIndex+ "⟨br/⟩ " + " 最后一个 ' s '的位置是 :"+lastIndex+
【"<br/>'"+"字符串对象的长度是:"+len+"<br/>'"+"数组arr中的各个元素为:"+"<br/>'"+arr[0]+"<br/>'"+arr[1]+
(br/)''+arr[2]+(br/)''+arr[3]+(br/)''+arr[4]
document.write(result+"\br/\\br/\");
</script>
```

3. Math 对象

• Math对象用于执行数学计算。Math对象和前面介绍的两类对象不同的是,在使用时不需要创建对象,而是直接使用Math来访问属性或方法,例如Math.PI。

• Math 对象常用属性:

- E: 欧拉常量,自然对数的底,约等于 2.7183

-PI: π,约等于 3.14159

Math 对象的常用方法

方法

atan(num)

abs(num)	返回num的绝对值
ceil(num)	返回大于等于num的最小整数
floor(num)	返回小于等于num的最小整数
max(n1,n2)	返回n1、n2中的最大值
min(n1,n2)	返回n1、n2中的最小值
pow(n1,n2)	返回n1的n2次方
sqrt(n)	返回n的平方根
random()	产生0-1之间的随机数
round(num)	返回num四舍五入后的整数
exp(num)和log(num)	返回以e为底的指数和自然对数值
sin(radianVal)、 cos(radianVal) 和 tan(radianVal)	分别是返回一个角的正弦、余弦和正切值的三角函数,方法输入参数以弧度表示
asin(num)、acos(num)和	分别反正弦、反余弦和正切三角函数,这

些函数的返回值以弧度表示

意 义

· Math对象属性和方法的使用:

Math.属性 Math.方法(参数1,参数2,…)

Math 对象

```
<!doctype html>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Math.row
Math.row
M
```

</script>

</head>

<body>

</body>

</html>

document.write("Math.max(2,3) = "+Math.max(2,3) + "
");//求两个值中的最大值document.write("Math.exp(0) = "+Math.exp(0) + "
");//求0的幂指数document.write("Math.sqrt(3) = "+Math.sqrt(3) + "
");//求3的平方根document.write("Math.pow(2,6) = "+Math.pow(2,6) + "
");//求2的6次幂document.write("Math.round(2.6) = "+Math.round(2.6) + "
");//四舍五入//求随机数,并对随机数四舍五入document.write("Math.round(Math.random()*10) = "+Math.round(Math.random()*10));

document.write("Math.ceil(15.6)="+Math.ceil(15.6)+"
");//求最高值

document.write("Math.floor(15.6)="+Math.floor(15.6)+"
");//求最低值

Math 对象示例

• 网页随机显示图片

4. Date 对象

- Date 对象可用来获取日期和时间
- 创建Date对象的方法:

var dt=new Date([日期参数])

• 日期参数说明:

- 省略不写,用于获取系统当前日期和时间 today=new Date()
- 日期字符串,其格式可以使用Date.parse()方法识别的任何一种,如,"October 1,2001"、"12/23/1999 12:06:36"等

time=new Date("12/23/1999 12:06:36 ") time=new Date("October 1,2001")

- 一律以数值表示,格式为: [公元年,月,日,时,分,秒]或简写成[公元年,月,日]

time=new Date(2001,10,10,0,0,0) time=new Date(2001,10,10)

· Date 对象的常用方法(1)

方法	意 义
getDate()	根据本地时间返回Date对象的日期1~31
getDay()	根据本地时间返回Date对象的星期数0~6
getMonth()	根据本地时间返回Date对象的月份数0~11
getYear()	根据本地时间,返回Date对象的年份数(在2000年以前返回年份数后两位,2000年以后返回4位)
getFullYear()	根据本地时间,返回以4位整数表示的Date对象年份数
getHours()	根据当地时间返回Date对象的小时数,24小时制
getMinutes()	根据当地时间返回Date对象的分钟数
getSeconds()	根据当地时间返回Date对象的秒数
getTime()	根据当地时间返回自1970年1月1日00:00:00以来的毫秒数
Date.parse(日期字符串)	根据当地时间返回自1970年1月1日00:00:00 以来的毫秒数

Date 对象的常用方法(2)

方法	意义
setYear(年份数)	根据本地时间设置Date对象的年份数
setFullYear(年份数[,月份,日期数])	根据本地时间设置Date对象的年份数
setDate(日期数)	根据本地时间设置Date对象的当月号数
setMonth(月[,日])	根据本地时间设置Date对象的月份数
setHours(小时[,分,秒, 毫秒])	根据本地时间设置Date对象的小时数
setMinutes(分[,秒,毫秒])	根据本地时间设置Date对象的分钟数
setSeconds(秒[,毫秒])	根据本地时间设置Date对象的秒数
setMilliSeconds(毫秒)	根据本地时间设置Date对象的毫秒数
setTime (总毫秒数)	根据本地时间设置Date对象自1970年1月1日00:00:00以来的毫秒数
toLocalString()	以本地时区格式显示,并以字符串表示

• Date对象属性和方法的使用:

Date对象.属性

Date对象.方法(参数1,参数2,…)

※例外:

Date.parse(日期字串)
Date.UTC()

Date 对象示例

```
<!doctype html>
<html>
<head>
                                                           0 +
<meta charset="utf-8">
<title>Date对象的使用</title>
                                                       Date对象的使用
<script type="text/javascript">
                                                       现在时间是: 2018年1月5日23:05:22 星期五
    var now = new Date(); //对系统当前时间创建Date对象
    var year=now.getFullYear(); //获取以四位数表示的年份
                                                       当前时间调用toLocaleString()的结果: 2018年1月5日 23:05:22
   var month=now.getMonth()+1; //获取月份
                                                       当前时间调用toString()的结果: Fri Jan 05 2018 23:05:22
                                                       GMT+0800 (中国标准时间)
    var date=now.getDate();//获取日期
                                                       当前时间调用toUTCString()的结果: Fri. 05 Jan 2018 15:05:22 GMT
    var day=now.getDay(); //获取星期数
    var hour=now.getHours();//获取小时数
    var minute=now.getMinutes();//获取分钟数
    var second=now.getSeconds();//获取秒数
    //创建星期数组
    var week=new Array("星期日","星期一","星期二","星期三","星期四","星期五","星期六");
    hour=(hour<10) ? "0"+hour:hour;//以两位数表示小时
    minute=(minute<10) ? "0"+minute:minute;//以两位数表示分钟
    second=(second<10) ? "0"+second:second;//以两位数表示秒数
    document.write("现在时间是: "+year+"年"+month+"月"+date+"目"+hour+":"+minute+
        ":"+second+" "+week[day]+"<hr>");
    document.write("当前时间调用toLocaleString()的结果: "+now.toLocaleString()+
        "<br>");
    document.write("当前时间调用toString()的结果: "+now.toString()+"<br>");
    document.write("当前时间调用toUTCString()的结果: "+now.toUTCString()+"<br>");
</script>
</head>
<body>
</body>
```

</html>

Date 对象示例

• 定时器