第十二章 JavaScript对象

Web程序设计基础

学习目标

什么是对象?

对象种类有哪些?

掌握字符串对象的定义及属性、方法的调用

掌握数学对象的定义及属性、方法的调用

掌握日期对象的定义及属性、方法的调用

掌握数组对象的定义及属性、方法的调用

目录

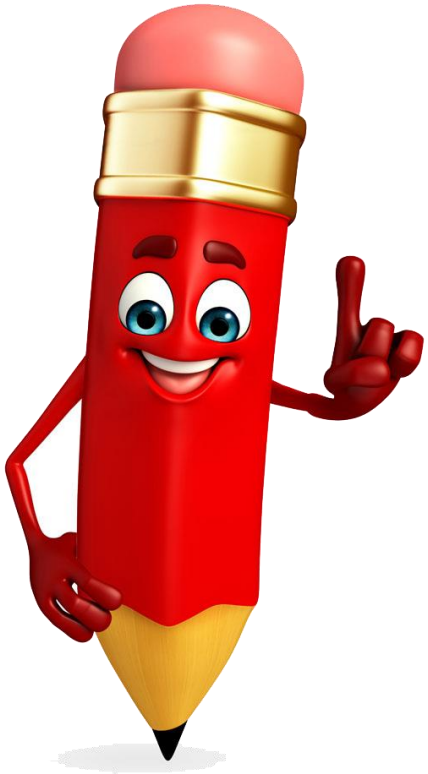
12.0 对象概述

1. 什么是对象

- “人”具有姓名、性别、年龄、身高、体重等特性
- “人”可以做吃饭、开车、运动等动作

12.0 对象概述

1. 什么是对象



- “一支笔”包括材质、型号等特性
- “一支笔”又包含画画、写字等动作

12.0 对象概述

1. 什么是对象

在计算机世界中的对象，不仅包括来自于客观世界的对象，还包含为解决问题而引入的抽象对象。例如，一个用户可以看做一个对象，它包含用户名、用户密码等特性，也包含注册、注销等动作。一个web页可以看做一个对象，它包含背景色、段落[illegible]标题等特性，同时又包含打开[illegible]

属性：
用来描述对象特性的数据，即若干变量。

方法：
用来操作对象的若干动作，即若干函数。

【结论】

对象就是一组属性与方法的集合。

2. 对象的种类

在JavaScript中对象分为内置对象、宿主对象和自定义对象三种。

- 内置对象是指系统预先定义好的，直接使用的对象。包括：字符串对象、数学对象、日期对象、数组对象等。
- 宿主对象是执行JS脚本的环境提供的对象，对于嵌入到网页中的JS来说，其宿主对象就是浏览器提供的对象，所以又称为浏览器对象。
- 自定义对象是开发人员自己定义的对象。

1. 字符串对象的定义

字符串对象是常用的对象之一，可提供诸多方法实现字符串检查、抽取、连接、分隔等字符串相关操作。

字符串对象定义1

[var] 字符串变量=字符串;

例如：var string1 = “Hello”;

字符串对象定义2

[var] 字符串对象=new String(字符串);

例如：var string1 = new String(“Hello”);

2. 字符串对象的属性

属　性	描　述
constructor	字符串对象的函数模型
length	字符串长度
prototype	添加字符串对象的属性

字符串对象的调用格式：

字符串对象属性使用

```
对象名.属性名                    //获得对象属性值
对象名.属性名 = 值    //为属性赋值
```

2. 字符串对象的属性

[点击查看案例]

```
<!doctype html>
<html>
 <head>
  <meta charset="utf-8">
  <title>字符串及属性调用</title>
  <script>
    var txt=“Hello World!”;
    document.write(txt.length);
  </script>
 </head>
 <body>
 </body>
</html>
```

3. 字符串对象的方法

方法	描 述
charAt(位置)	字符串对象在指定位置处的字符
charCodeAt(位置)	字符串对象在指定位置处字符的Unicode值
indexOf(要查找的字符串,[起始位置])	返回被查找的字符串在原字符串对象中首次出现的位置
lastIndexOf(要查找的字符串)	返回被查找的字符串在原字符串对象中最后出现的位置，从后往前进行查找
substr(开始位置,[长度])	从字符串对象指定位置开始，按照指定的数量截取字符，并返回截取的字符串
substring(开始位置，结束位置)	从字符串对象指定的位置开始，截取到结束位置，并返回截取的字符串
split([分隔符])	分割字符串到一个数组中
replace(要查找的字符串,新字符串)	在字符串对象中，将指定的字符串替换为新字符串
toLowerCase()	字符串对象转换为小写
toUpperCase()	字符串对象转换为大写

3. 字符串对象方法的调用

字符串对象方法调用

字符串对象.字符串对象方法

【例12-1】设计程序，在文本框输入字符串，单击【检查】按钮，检查字符串是否为有效字符串。约定字符串只能由大小写字母、数字、下划线“_”和连字符（减号）“-”构成。如果输入的字符串有效，弹出对话框“合法字符串。如果输入的字符串无效，弹出对话框“不合法字符串”。

[点击查看案例]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>检验字符串是否合法</title>
<script>
function inputCheck(userStr)
{
    var
charSet="ABCDEFGHIJKLMNOPQRS
TUVWXYZabcdefghijklmnopqrstuvw
xyz1234567890_-";
    for(i=0;i<userStr.length;i++)
   {
 if(charSet.indexOf(userStr.charAt(i))
== -1)
    {alert("不合法字符串");
          return;}
    }
          alert("合法字符串");
}
</script>
</head>
<body>
<form id="myform"
name="myform" method="post">
  <input type="text"
name="txtString" id="txtString">
  <input type="button"
name="button" id="button"
value="检 查"
onClick="inputCheck(document.myf
orm.txtString.value)">
</form>
</body>
</html>
```

12.2 数学对象

数学对象也是常用的对象之一，提供了一些数学运算中的**常数**及数学**计算方法**，在数学运算时非常有用。数学对象**无需创建**，通过把Math作为对象使用就可以调用其所有属性和方法。

例如：

```
var fPi = Math.PI;

document.write(fPi);
```

1. 数学对象的属性

属性	描 述
E	返回算术常量e，即自然对数的底数（约等于2.718）
LN2	返回2的自然对数（约等于0.693）
LN10	返回10的自然对数（约等于2.302）
LOG2E	返回以2为底的e的对数（约等于1.442）
LOG10E	返回以10为底的e的对数（约等于0.434）
PI	返回圆周率（约等于3.14159）
SORT2	返回2的平方根（约等于1.414）
SQRT1_2	返回2的平方根的倒数（约等于0.7071）

2. 数学对象的方法

方法	描 述	方法	描 述
abs(x)	返回数字的绝对值	min(x,y)	求最小值
ceil(x)	对数字进行上取整	pow(x,y)	返回x的y次幂
floor(x)	对数字进行下取整	sqrt(x)	返回数字的平方根
round(x)	对数字进行四舍五入	random()	返回0~1之间的随机数
exp(x)	返回e的指数	sin(x)/cos(x)/tan(x)	计算x的正弦、余弦、正切值
log(x)	返回数字的自然对数（底为e）	asin(x)/acos(x)/atan(x)	计算x的反正弦、反余弦、反正切值
max(x,y)	求最大值		

2. 数学对象的方法

Javascript针对数值（Number）对象提供了toFixed方法和toPrecision方法，可实现对数值型小数位数的保留操作。

方法	描 述
toFixed(x)	返回某数四舍五入之后保留x位小数
toPrecision(x)	返回某数四舍五入之后保留x位字符

方法调用

```
数字. toFixed(x);       //保留x位小数
数字.toPrecision(x);    //保留x位数字
```

12.2 数学对象

【例12-2】设计程序，单击【随机数】按钮，使用Math对象的random方法产生一个0~100之间（含0和100）的随机整数，并通过消息对话框显示该随机数。单击【计算】按钮，计算该随机数的平方、平方根和自然对数，保留2位小数，并在消息对话框中显示。

算法分析：

- random()方法可产生一个0~1的随机数，将这个数乘以100，可获得0~100的随机数。
- 对0~100的随机数用floor(x)方法对下取整，即满足取数要求。
- 用pow(x,y)方法计算随机数的平方
- 用sqrt(x)方法计算随机数的平方根
- 用log(x)方法计算随机数的自然对数
- 用toFixed(x)方法对计算结果保留2位小数

例题12-2程序实现

[点击查看案例]

```
<!doctype html>
<html>
 <head>
 <meta charset="utf-8">
 <title>产生随机整数，计算平方、平方根和自然对数</title>
 <script>
   var data;
   function getRandom(){
      data=Math.floor(Math.random()*100);
      alert("随机整数为： "+ data);}
   function calc(){
      var a = Math.pow(data,2);
      var b = Math.sqrt(data).toFixed(2);
      var c = Math.log(data).toFixed(2);
     alert("随机整数"+data+"计算结果为： \n平方\t平方根\t自然对数
\n"+a+"\t"+b+"\t"+c);}
 </script> </head>
```

例题12-2程序实现

[点击查看案例]

```
<body>
<form id="myForm" name="myForm" method="post">
  <input type="button" value="随机数" onClick="getRandom()">
  <input type="button" value="计 算" onClick="calc()">
</form>
</body>
</html>
```

12.3 日期对象

JavaScript脚本内置了本地对象Date，该对象可以表示从毫秒到年的所有时间和日期，并提供了一系列操作时间和日期的方法。

1. 创建日期对象

日期对象定义1

获取本地时间

```
new Date();
```

例如：

```
var date1 = new Date();
alert(date1);
```

12.3 日期对象

1. 创建日期对象

创建指定的时间对象

日期对象定义2

new Date(日期字符串);

例如：

```
var date2=new Date("2016/8/1 8:30:30");
alert(date2);
```

返回结果：

Mon Aug 01 2016 80:30:30 GMT+0800

12.3 日期对象

1. 创建日期对象

多参数创建指定的时间对象

日期对象定义3

new Date(年,月,日,[时,分,秒,毫秒]);

例如：

```
var date3=new Date(2016,9,1,5,30,30);
alert(date3);
```

返回结果：

Thu Sep 01 2016 05:30:30 GMT+0800

1. 创建日期对象

传递一个毫秒数值来创建指定的时间对象

日期对象定义4

```
new Date(毫秒);
```

例如：

```
var date4=new Date(1000000000000);
alert(date4);
```

返回结果：

Sun Sep 09 2001 09:46:40 GMT+0800

注意：计算起点为1970年1月1日0时0分0秒0毫秒

【例12-3】使用四种方法创建日期对象。

[点击查看案例]

```
!doctype html>
<html><head>
 <meta charset="utf-8">
 <title>创建日期对象</title>
  <script>
    var myDate1=new Date();
    var myDate2=new Date("June 10,2018");
    var myDate3=new Date("2018/10/1");
    var myDate4=new Date(2018,10,19,16,15,14);
    var myDate5=new Date(20000);
    document.write("myDate1时间为：  "+myDate1.toLocaleString()+"<br>");
    document.write("myDate2时间为：  "+myDate2.toLocaleString()+"<br>");
    document.write("myDate3时间为：  "+myDate3.toLocaleString()+"<br>");
    document.write("myDate4时间为：  "+myDate4.toLocaleString()+"<br>");
    document.write("myDate5时间为：  "+myDate5.toLocaleString()+"<br>");
  </script>  </head>
 <body></body>
</html>
```

2. 日期对象的方法

日期对象提供了许多操作日期和时间的**方法**，在脚本开发过程中可简单快速的控制日期和时间。

日期对象方法调用

```
日期对象.日期对象方法
```

2. 日期对象的方法

➢ 常用日期及时间的获取

方法	描 述
Date()	返回当日的日期和时间
getDate()	从 Date 对象返回一个月中的某一天 (1 ~ 31)
getDay()	从 Date 对象返回一周中的某一天 (0 ~ 6)
getMonth()	从 Date 对象返回月份 (0 ~ 11)
getFullYear()	从 Date 对象以四位数字返回年份
getYear()	请使用 getFullYear() 方法代替
getHours()	返回 Date 对象的小时 (0 ~ 23)
getMinutes()	返回 Date 对象的分钟 (0 ~ 59)
getSeconds()	返回 Date 对象的秒数 (0 ~ 59)
getMilliseconds()	返回 Date 对象的毫秒(0 ~ 999)
getTime()	返回 1970 年 1 月 1 日至今的毫秒数

当前日期显示

[点击查看案例]

```
<!DOCTYPE html>
<html>
 <head>
   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
   <title>Date对象的应用1</title>
 </head>
 <body>
  <script type="text/javascript">
    var date=new Date();
    var year=date.getFullYear();
    var month=date.getMonth();
    month=month+1;
    var day=date.getDate();
    document.write("当前日期为:"+year+"年"+month+"月"+day+"日");
  </script>
 </body>
</html>
```

2. 日期对象的方法

➢ 常用日期及时间的设置

方法	描 述
setDate()	设置 Date 对象中月的某一天 (1 ~ 31)
setMonth()	设置 Date 对象中月份 (0 ~ 11)
setFullYear()	设置 Date 对象中的年份（四位数字）
setYear()	请使用 setFullYear() 方法代替
setHours()	设置 Date 对象中的小时 (0 ~ 23)
setMinutes()	设置 Date 对象中的分钟 (0 ~ 59)
setSeconds()	设置 Date 对象中的秒钟 (0 ~ 59)
setMilliseconds()	设置 Date 对象中的毫秒 (0 ~ 999)
setTime()	以毫秒设置 Date 对象
SetTimeout	

[点击查看案例]

```
<!DOCTYPE html>
<html>  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Date对象的应用2</title>  </head>
 <body>
  <script type="text/javascript">
     var date=new Date();
     date.setFullYear(2020,10,25);
     var year=date.getFullYear();
     var month=date.getMonth();
     var day=date.getDate();
     document.write("设置日期为:"+year+"年"+month+"月"+day+"日");
     date.setDate(date.getDate()+2);
     document.write(“<br/>更改日期为:"+date);
  </script>
 </body></html>
```

【例12-5】计算当前日期距离元旦（1月1日）还有多久？分别用：毫秒、秒、分钟、小时、天显示结果。

[点击查看案例]

```
<!doctype html>
<html><head>
<meta charset="utf-8">
<title>日期相减</title>
<script>
var now=new Date();
var nationalDay=new Date(2022,0,1,0,0,0);
var msel=nationalDay-now;
document.write("距离2021年元旦还有： "+msel+"毫秒<br>");
document.write("距离2021年元旦还有： "+parseInt(msel/1000)+"秒<br>");
document.write("距离2021年元旦还有： "+parseInt(msel/(60*1000))+"分钟<br>");
document.write("距离2021年元旦还有： "+parseInt(msel/(60*60*1000))+"小时
<br>");
document.write("距离2021年元旦还有： "+parseInt(msel/(60*60*24*1000))+"天
<br>");
</script></head>
<body></body>
</html>
```

2. 日期对象的方法

➢ window对象中和时间有关的方法

①SetTimeout： 用于设置一个计时器，在指定的时间间隔后调用函数或计算表达式，且仅执行一次。

基本格式

```
SetTimeout（函数名，间隔时间-毫秒）
```

② clearTimeOut： 清除由setTimeout()方法所设置的计时器。

基本格式

```
clearTimeout(setTimeout() 返回的 ID 值)
```

计时器设计案例

[点击查看案例]

漂亮的计时器

```
<!doctype html>
<html><head>
 <script type="text/javascript">
  var timeResult=0
  var timer
  function timedCount()
  {  document.txtForm.txt.value=timeResult;
  timeResult=timeResult+1
  timer=setTimeout("timedCount()",1000)}
  function stopCount()
  {clearTimeout(timer)}
 </script> </head>
<body>
<form id="txtForm" name="txtForm" method="get">
<input type="button" value="开始！ " onClick="timedCount()">
<input type="text" name="txt" id="txt">
<input type="button" value="停止！ " onClick="stopCount()">
</form></body></html>
```

2. 日期对象的方法

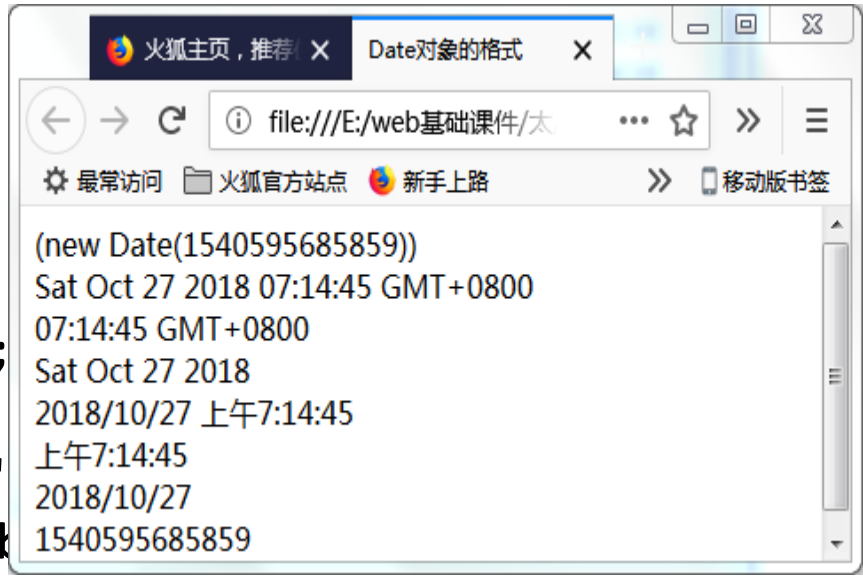
- 日期对象的格式转换

方法	描 述
toString()	把 Date 对象转换为字符串
toTimeString()	把 Date 对象的时间部分转换为字符串
toDateString()	把 Date 对象的日期部分转换为字符串
toLocaleString()	根据本地时间格式，把 Date 对象转换为字符串
toLocaleTimeString()	根据本地时间格式，把 Date 对象的时间部分转换为字符串
toLocaleDateString()	根据本地时间格式，把 Date 对象的日期部分转换为字符串
valueOf()	返回 Date 对象的原始值

日期对象的格式案例

[点击查看案例]

```
<!DOCTYPE html>
<html>  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Date对象的应用2</title>  </head>
 <body>
   <script type="text/javascript">
     var d = new Date();
     document.write(d.toString()+"<br/>");
     document.write(d.toTimeString()+"<br/>");
     document.write(d.toDateString()+"<br/>");
     document.write(d.toLocaleString()+"<br/>"
     document.write(d.toLocaleTimeString()+"<b
     document.write(d.toLocaleDateString()+"<br/>");
     document.write(d.valueOf()+"<br/>");
   </script>
</head>
<body></body>
</html>
```



12.4 数组对象

数组对象是一个有序的数据集合，使用单独的变量名来存储一系列的数据。在JavaScript中，定义数组时不需要指定数组的数据类型，而且可以将不同类型的数据存放到一个数组中。

1. 数组对象的创建

JavaScript中使用**new**关键字创建**Array**对象，创建数组对象有三种方法。

- 空数组　　var 数组名=new Array();

例如，声明数组a，长度为0，代码如下：

```
var a=new Array();
```

1. 数组对象的创建

- 声明长度为n的空数组　　var 数组名=new Array(n);

例如，声明数组b，长度为10，代码如下：

```
var b=new Array(10);
```

- 通过参数列表赋值，创建数组

var 数组名=new Array(元素1,元素2,元素3,…);

例如，声明数组c，并为元素赋值为1，2，3，4，代码如下：

```
var c=new Array(1,2,3,4);
```

2. 数组对象的操作

数组的长度属性

数组对象的属性非常少，最常用的属性为**length**，可以返回数组对象的长度，也就是数组中元素的个数。

[点击查看案例]

```
<!DOCTYPE html>
<html> <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>数组对象的属性</title> </head>
 <body>
   <script type="text/javascript">
     var myArray=new Array("a","b","c");
     document.write("数组长度为： "+myArray.length);
   </script></head>
<body></body>
</html>
```

2. 数组对象的操作

访问数组

引用数组元素是通过数组的**序列号**。在JavaScript数组中的元素序列号是从**0**开始计算的，然后依次加**1**。

对数组赋值的格式

```
数组变量[i]=值;
```

由数组取值的格式

```
变量名=数组变量[i];
```

12.4 数组对象

遍历数组元素

```
for (变量 in 数组对象)
{
  在此执行代码
}
```

[点击查看案例]

```
<html><head></head>
<body>
<script type="text/javascript">
var x
var mycars = new Array()
mycars[0] = "Saab"
mycars[1] = "Volvo"
mycars[2] = "BMW“
for (x in mycars){document.write(mycars[x] + "<br />“)}
</script>
</body></html>
```

2. 数组对象的操作

【例12-6】创建长度为3的数组myArray，并且对第1个元素赋值，然后分别输出三个元素。

[点击查看案例]

```
<!doctype html>
<html><head>
<meta charset="utf-8">
<title>访问数组</title>
<script>
var myArray= new Array(3);
myArray[0]=10;
document.write("元素1的值为： "+myArray[0]+"<br>元素2的值为：
"+myArray[1]+"<br>元素3的值为： "+myArray[2]);
</script>
</head>
<body></body>
</html>
```

2. 数组对象的操作

添加数组元素

- 假设现有数组长度为3，通过修改length属性为5，如

```
myArray.length=5;
```

这时数组增加2个元素。新增的2个元素值为undefined。

- 现有数组a，长度为3，该数组包含的元素是a[0]、a[1]、a[2]。如果添加数组赋值语句

```
a[4]=8;
```

这时数组增加2个元素。其中a[3]的值为undefined，a[4]的值为8。

2. 数组对象的操作

删除数组元素

- 假设现有数组长度为5，通过修改length属性为3，如

```
myArray.length=3;
```

这时数组将删除尾部的2个元素。

- 使用delete元素将数组中某一元素取值删除，如

```
delete myArray[1];
```

这时数组长度不变，其中myArray [1]的值为undefined。

2. 数组对象的操作

[点击查看案例]

```
<!doctype html>
<html><head>
<meta charset="utf-8">
<title>数组的删除</title>
<script>
 var myArray2=new Array("a","b","c","d","e");
 myArray2.length=3;
 document.write(myArray2.length+"<br/>");
 delete myArray2[1];
 document.write(myArray2.length+"<br/>");
 document.write(myArray2[1]);
</script>
</head>
<body></body>
</html>
```

3. 数组对象的方法

方法	描 述
concat(数组1,数组2,...)	合并数组
join(分隔符)	将数组转换为字符串
pop()	删除最后一个元素，返回最后一个元素
push(元素1,元素2,...)	添加元素，返回数组的长度
shift()	删除第一个元素，返回第一个元素
unshift(元素1,元素2,...)	添加元素至数组开始处
slice(开始位置[,结束位置])	从数组中选择元素组成新数组
splice(位置,多少[,元素1,元素2,...])	从数组中删除或替换元素
sort()	排序数组
reverse()	倒排序数组
toString	返回一个字符串，该字符串包含数组中所有元素，各元素间用逗号分隔

3. 数组对象的方法

□ 数组合并及数组元素的增加、删除

【例12-7】新建数组a并赋值“A”，“S”，“D”，新建数组b并赋值“X”，“Y”，“Z”，新建数组c、d、e进行如下操作：

1）将数组a和b合并到数组c并输出数组c。

2）删除数组c的第一个元素和最后一个元素并输出数组c。

3）在数组c尾部增加三个元素“b”，“n”，“m”，输出数组c。

4）在数组c开头增加三个元素“q”，“w”，“e”，输出数组c。

5）截取数组c下标2到下标4的元素，并赋值给数组d，输出新数组d。

6）将数组d转换为字符串s1，使用空格作为分隔符，输出字符串s1。

7）创建字符串s2并赋值“北京，上海，广州”，以逗号为分隔符，将字符串转到数组e，输出数组e。

数组对象的方法应用案例1

方法要点：用 concat方法来合并数组；用pop方法和shift方法可以删除数组元素； 用push方法和unshift方法可以增加数组元素，用slice 方法选择并组成新数组。

[点击查看案例]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>数组合并及元素增删</title>
<script src="1.js">
</script>
</head>
<body>
</body>
</html>
```

脚本文件

3. 数组对象的方法

排序数组和反转数组

【例12-8】新建数组x并赋值40,100,1,5,25,10，使用sort方法排序数组，并输出数组x到页面。

[点击查看案例]

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>数组排序</title>
<script src="1.js">
</script>
</head>
<body>
</body>
</html>
```

脚本文件

本章结束