

Javascript

马凡 李蟠 编著

实例
应用



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

00102435

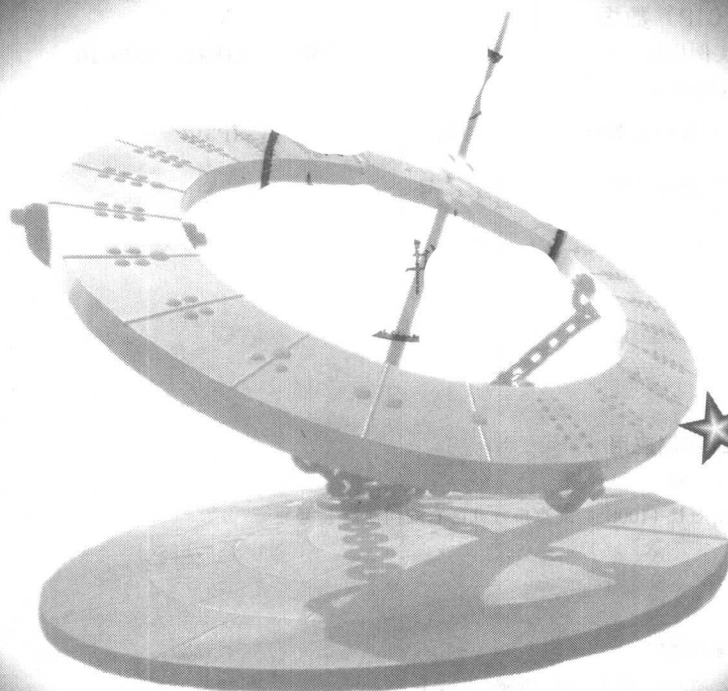
TP312JA

81

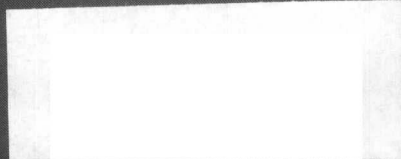


JavaScript 实例应用

马 凡 编著
李 塔



中国铁道出版社



(京)新登字 063 号

内 容 简 介

本书主要以实例为主,全书共分4章,第1章主要介绍了 JavaScript 脚本语言的主要特征和基本功能,并通过一些实例来帮助读者加深理解;第2章主要介绍了编写 JavaScript 脚本的规则以及如何在 HTML 文档中嵌入 JavaScript 脚本语言;第3章主要介绍了 JavaScript 系统中预先定义的对象,相比用户自定义的对象而言,预定义对象的利用率更高;第4章给出了大量的精彩实用范例,除了可供练习之用以外,读者还可以在实际应用中稍加修改,即可运用到自己的网页中。

本书特别适用于初、中级网页编程人员,但对高级网页编程人员也是不可缺少的参考手册。

图书在版编目(CIP)数据

JavaScript 实例应用/马凡,李蟠编著. —北京:中国铁道出版社,2000.10

ISBN 7-113-03945-6

I. J… II. ①马…②李… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 54536 号

书 名: JavaScript 实例应用

作 者: 马 凡 李 蟠

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟

特邀编辑: 王占清

封面设计: 冯龙彬

印 刷: 北京市兴顺印刷厂

开 本: 787×1092 1/16 印张: 16 字数: 386 千

版 本: 2001 年 1 月第 1 版 2001 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-03945-6/TP·481

定 价: 26.00 元

版权所有 盗印必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

目 录

第 1 章 JavaScript 概述	1
1.1 JavaScript 与 Java	1
1.2 JavaScript 初探	2
1.2.1 实例一	2
1.2.2 实例二	3
1.2.3 实例三	4
1.2.4 实例四	5
1.2.5 实例五	6
1.2.6 实例六	7
1.2.7 实例七	9
1.2.8 实例八	10
第 2 章 JavaScript 语言基础	12
2.1 对象的概念	12
2.1.1 JavaScript 对象	13
2.1.2 JavaScript 的对象属性	13
2.1.3 预定义对象	15
2.2 方法的概念	19
2.3 事件的概念	22
2.4 变量的概念	23
2.5 程序控制结构	24
2.5.1 JavaScript 语句	24
2.5.2 For 语句	24
2.5.3 While 语句	25
2.5.4 Do-While 语句	26
第 3 章 JavaScript 预定义对象	33
3.1 Windows 对象	33
3.1.1 Windows 对象说明	33
3.1.2 Windows 对象的属性、方法和事件	35



3.1.3 打开或关闭一个窗口	35
3.2 Frame 对象	43
3.2.1 帧的概念	43
3.2.2 帧组和帧的属性	44
3.2.3 利用 Frame 编制程序	44
3.3 Document 对象	47
3.3.1 Document 对象的属性和方法	47
3.3.2 使用 Document 对象	48
3.3.3 综合实例	49
3.4 Form 对象	56
3.4.1 文本框	56
3.4.2 按钮	61
3.5 复位按钮和提交按钮	71
3.5.1 复选按钮的属性和方法	71
3.5.2 使用提交按钮和复位按钮	71
3.6 复选框和单选按钮	75
3.6.1 复选框的属性、方法	75
3.6.2 单选框	77
3.7 窗体的事件处理	82
3.7.1 窗体事件处理器	83
3.7.2 本地窗体处理	83
3.7.3 向 CGI 程序发送数据	83
3.8 Cookie 的使用和隐藏字段	86
3.8.1 Cookie 与隐藏字段的比较	86
3.8.2 隐藏字段的使用	87
3.8.3 Cookie 概念	94
3.9 JavaScript 图像处理	98
3.9.1 由按钮控制的图形	98
3.9.2 使用 Image 对象和 Images[] 数组	100
3.9.3 Internet Explorer 中的结构化控件	109
3.10 JavaScript 算术运算	121
3.10.1 算术函数的基本概念	121
3.10.2 JavaScript 提供的算术常数	122
3.10.3 Math 对象提供的算术函数	122
3.10.4 算术函数的用法	123
3.11 浏览器、地址和历史对象	127
3.11.1 浏览器对象	127
3.11.2 使用地址对象	129
3.11.3 使用历史对象	131

第 4 章 JavaScript 开发综合实例	136
4.1 禁止使用鼠标右键	136
4.2 自动滚屏	137
4.3 图片的分割	139
4.4 带链接的跑马灯	140
4.5 状态栏跑马灯	141
4.6 简单计算	142
4.7 字符连续滚动显示	143
4.8 带链接的文本垂直滚动显示	145
4.9 带链接的文本水平滚动显示	147
4.10 飞行的图片	149
4.11 始终指向选择内容的手	151
4.12 显示器分辨率检测	155
4.13 状态栏动态信息	156
4.14 自定义滚动信息框	158
4.15 多条消息随机显示	161
4.16 鼠标事件驱动的不同图片的显示	163
4.17 用按钮控制的弹出消息框	165
4.18 用单选按钮控制的弹出消息框	166
4.19 连续显示多个消息框	167
4.20 鼠标光标移进时弹出消息框	168
4.21 可折叠打开的导航菜单	169
4.22 选择背景主题	173
4.23 随机颜色	175
4.24 网页编辑器	176
4.25 可控时钟	177
4.26 状态栏时钟	179
4.27 模拟跳舞	179
4.28 显示当前时段	184
4.29 检测视频插件	186
4.30 2001 年倒计时	187
4.31 字符串加密	191
4.32 Cookie 日历	193
4.33 设计一个日历	201
附录 JavaScript 语言参考	207

第 1 章 JavaScript 概述

本章从 JavaScript 的由来谈起,介绍了该脚本语言的主要特征和基本功能,并通过一些实例来帮助读者加深理解。

1.1 JavaScript 与 Java

讲到 JavaScript 语言,就不能不讲 Java 语言,下面向读者简单介绍一下这两种语言的由来。

Java 语言的产生始于 1991 年,当时万维网尚处于起步阶段,以生产工作站起家的 Sun 公司试图寻求其他领域的发展,把目光投向消费类电子产品,如电子游戏机、个人数字助理等。为此,需要创建一种软件平台,并且要求具有平台无关性。

最初,Java 的主要开发者之一 James Gosling 尝试采用 C++,但 C++太复杂、安全性差,开发组成员最后决定重新开发一种新的语言,以 C++为基础,吸收 C++面向对象技术的优点,剔除 C++内存管理、指针等不安全因素。经过一年多的开发,Java 的前身——Oak 出现了。但事与愿违,Sun 公司向消费类电子产品进军的构想一再受挫。

随着万维网的盛行,开发小组突然发现,Oak 的主要特征——与平台无关、系统安全性和可靠性一旦与万维网结合,将具有广阔的前景。在经过一番努力之后,Java 终于出现了。

1995 年初,发行万维网浏览器的 Netscape 公司宣布在其 Navigator 2.0 中率先支持 Java,从此,Java 语言受到万众瞩目,成千上万个万维网网站上出现了活泼可爱的 Java Applet (Java 小应用程序),使得 Web 页面一改往日静态的形象。随后各大公司相继宣布支持 Java 语言,这些公司包括:IBM、Microsoft、Intel、Oracle 等业界巨头。

用户通过 Java Applet 小应用程序,可以创建集图像、动画、声音、影像于一体的多媒体交互网站。

与 Java 程序设计语言不同,JavaScript 是一种描述性脚本语言。由于设计 Java Applet 小应用程序具有相当的难度,并不适合所有的网页设计者,Netscape 公司推出了另一种提高 Web 页面可交互性的方法——JavaScript。JavaScript 脚本可以被嵌入 HTML 文件之中,无须经过编译即可在浏览器中运行,Navigator 3.0 和 Internet Explorer 3.0 及以上版本都支持 JavaScript 脚本语言。

在 Web 页面中使用 JavaScript 脚本,可以做到:

- (1) 生成表单。
- (2) 设计动态 Web 页面。
- (3) 嵌入计算器。
- (4) 生成小游戏。
- (5) 向 Web 页面增加一个随机数的轮盘,在访问者按一个按钮时转动,然后根据它停住时指示的号码访问喜欢的站点。
- (6) 构造联机表格。



- (7) 构造多媒体播放设备的前面板。
- (8) 使用多帧结构。
- (9) 设计说明如何使用像 JavaScript 这样的工具编写脚本的在线帮助。

JavaScript 和 Java 很类似，但又不一样。Java 是一种比 JavaScript 复杂得多的程序设计语言，而 JavaScript 则比 Java 容易掌握。JavaScript 脚本不必经过编译即可在浏览器中运行，而使用 Java 程序设计语言编写的代码必须经过编译方可在浏览器中运行。实际上，之所以提倡在网页中使用 JavaScript 而避免使用 Java，就是因为 JavaScript 脚本可以在浏览器端直接运行，并且 JavaScript 脚本短小精悍，有助于提高网页的交互性能和浏览速度。

需要注意的是：

- ☛ 不支持 JavaScript 语言的浏览器会将 JavaScript 代码当作页面内容的一部分，从而导致页面显示混乱。因此使用注释标识符 `<!-- 注释 -->` 围绕整个脚本代码就非常重要，这样做至少可以使不支持脚本语言的浏览器能正确地显示网页。
- ☛ 与 HTML 语言不同，脚本语言不使用 `<!-- 注释 -->` 作为注释，而使用 `//` 符号来标识脚本代码中的注释。任何 `//` 符号后的脚本代码都被当作是注释而不执行。
- ☛ `<script>` 标识符中的 `language` 属性并非必需，而且它的使用越来越少。
- ☛ 脚本代码通常放在 HTML 文档头部分，如 `<head></head>` 标识符之间，而不是放在文档体 `<body></body>` 标识符之间。这并不是严格的要求。事实上，脚本代码可以放在 HTML 文档的任何部分。但是为了结构明晰、便于维护，通常把不直接对文档正文产生影响的脚本代码放在文档头中。
- ☛ JavaScript 对变量名区分大小写，即 `Number` 和 `number` 是两个不同的变量。

1.2 JavaScript 初探

使用 JavaScript 脚本语言开发动态网页，仅需一个文本编辑器即可，如 Windows 环境下的记事本程序。那么，执行 JavaScript 脚本语言需要什么样的环境呢？根据我们的测试，Netscape 2.0 beta 3 版以上都支持 JavaScript 的功能，其他浏览器如 Internet Explorer 3.0 及以上版本当然也支持。

本节以一些例子来介绍如何将 JavaScript 放入 HTML 文档中，体会一下该脚本语言的特性。

需要注意的是，执行 JavaScript 脚本必须具备 Microsoft Internet Explorer 3.0 (或更高版本) 的环境。本节的全部实例均在 Internet Explorer 4.0 环境下测试通过，当然，所有的实例亦可在 Netscape Navigator 2.0 (或更高版本) 环境下执行。

1.2.1 实例一

下面介绍如何通过 JavaScript 在 Web 页面中显示一串文字。

```
<html>
```

```
<head>
```

```
My first JavaScript!
```



```
</head>
<body>
<br>
This is a normal HTML document.
<br>
<script language="JavaScript">
    document.write ("这是以 JavaScript 显示的!")
</script>
<br>
Back in HTML again.
</body>
</html>
```

注意：全书所有代码请到 www.bookposter.com/download.htm 下载。
执行结果如图 1-1 所示。



图 1-1 显示文字

读者可以了解到如何在 HTML 文件中使用<script>标记：这个标记可以放在文件的任何地方。

1.2.2 实例二

接下来这个例子所要介绍的是函数（function）的使用。通常，函数是在 HTML 文件的 body 部分被引用，而函数的定义最好放在 HTML 文件的相应部分，以便在 body 部分引用该函数时，它能被读取出来。另外，Script 标记的有关描述语法部分，可以用注解的符号将它括起来，以免无法读取 JavaScript 的浏览器读到而出错。

```
<html>
<head>
<script language="JavaScript">
    function pushbutton() {
        alert ("嗨！好")；
```



```
}  
</script>  
</head>  
<body>  
<form>  
  <input type="button" name="Button1" value="Push me" onclick="pushbutton()">  
</form>  
</body>  
</html>
```

执行结果如图 1-2 所示,当用鼠标单击按钮的时候,应该会出现一个对话框,上面有“嗨!好”的字串,这个结果是如何产生的呢?

首先,在 head 内的函数被载入内存中,接着一个 form 标记 `input type = "button"` 将产生一个按钮。然后,可以在后面看到 `onClick` 的指令,这就是告诉浏览器,当该按钮被按时,应会执行 `onClick` 后的函数 `pushbutton()`,而这个函数在脚本被载入时就已安放在内存中了。需要注意的是,在这个函数中使用了 `alert` 方法,这是 JavaScript 事先定义好的方法,它会以对话框的形式显示内涵的信息。在 JavaScript 脚本语言中,预定义了许多方法,读者可以参考本书的附录。

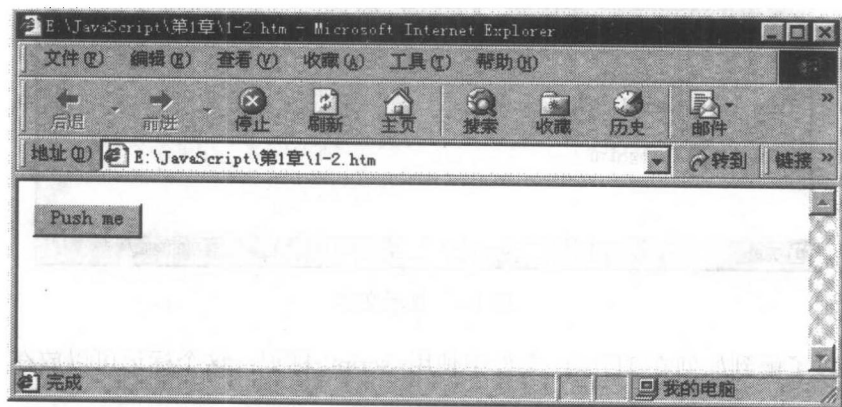


图 1-2 按钮

1.2.3 实例三

本实例将告诉读者如何通过一个文本框读入访问者输入的数据,事实上,使用一个函数就可以实现这个功能。

```
<html>  
<head>  
<script language="JavaScript">  
<!-- hide script from old browsers  
  function getname (str) {
```

```
        alert ("哈罗! "+ str+"!");  
    }  
    // end hiding contents -->  
</script>  
</head>  
<body>  
    请输入你的姓名:  
<form>  
        <input type="text" name="name" onBlur="getname (this.value) " value="">  
</form>  
</body>  
</html>
```

执行结果如图 1-3 所示。

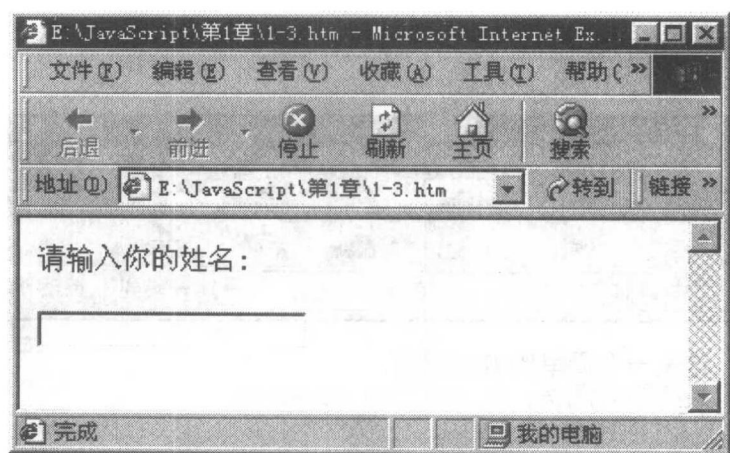


图 1-3 使用表单（文本框）

在这个例子中又有新的东西了。首先，注意一下注解部分的语法（<!-- ... -->），此部分应注意避免旧版本或是不支持 JavaScript 的浏览器因为不认识这些函数而产生错误。它的顺序应该是<script>在先，接着是注解的开头<!--，然后是内容，注解尾-->，最后是</script>。另外要注意的一点是，注解尾一行开头的双斜线“//”不可以省略，它代表了 JavaScript 的注解，若省略了的话，-->之前的文字会被误认为是 JavaScript 的指令。

这个例子可以让访问者输入一段文字，输入完毕后经由<input>标记中的 onBlur 事件函数测知，于是调用 Getname(Str)函数来取得输入字符串，并将它显示在对话框上。函数 Getname (this.value) 中的 this.value 是在文字输入格式中所输入的值。

1.2.4 实例四

通常制作完 HTML 文档后，可手工加上文件最后修改的日期，现在可以不用担心忘记了。编写一个如下的代码，就可以在每次修改文件以后自动产生最后修改时的日期了。

```
<html>
```



```
<head>
<title>最后修改的时间</title>
</head>
<body>
这是一个简单的 HTML 网页.
<br>
最终的时间是:
<script language="JavaScript">
<!-- hide script from old browsers
    document.write ( document.lastModified )
// end hiding contents -->
</script>
</body>
</html>
```

执行结果如图 1-4 所示。



图 1-4 动态显示时间信息

在本实例中, `document.lastModified` 参数在 Netscape 2.0 beta 2 版时要写成 `document.lastmodified`, 其后的版本就改为 `document.lastModified`。JavaScript 是区分大小写的, `lastmodified` 与 `lastModified` 在它看来是不同的结果。

1.2.5 实例五

通常将鼠标移到超链接上面时, 浏览器下的状态栏将发生相应的变化。在本实例中, 可以把这样的功能与 JavaScript 的功能相结合。这里, 把鼠标移到不同超链接上面, 将会有一个窗口出现。

```
<html>
<head>
```

```
<script language="JavaScript">

<!-- Hiding
    function hello() {
        alert ("哈 罗!");
    }

</script>
</head>
<body>
<a href="" onMouseOver="hello()">链接</a>
</body>
</html>
```

图 1-5 为执行结果。



图 1-5 链接显示信息

只要在<a>标记中加入 onMouseOver 方法,就可达成这种效果了。这里的 window.status 是用来在浏览器的状态栏上显示一些信息用的。可以看到,信息部分是用“”括起来的,而非以“”括起来,在信息部分结束之后,必须加上 return true。

1.2.6 实例六

这是一个使用日期和时间的例子。在前面曾经介绍了 lastModified 的用法和作法。现在要显示出的并不是网络上服务器或文件修改日期,而是客户端系统的日期和时间。

```
<html>
<head>
<title>
```



```
Hello
</title>
<script language="JavaScript">
<!-- Hiding
    today = new Date()
    document.write ("现在是: ", today.getHours(), ":", today.getMinutes())
    document.write ("<br>今天日期为: ",
        today.getMonth()+1, "/", today.getDate(), "/", today.getYear());
// end hiding contents -->
</script>
</head>
<body>
</body>
</html>
```

图 1-6 为运行结果。

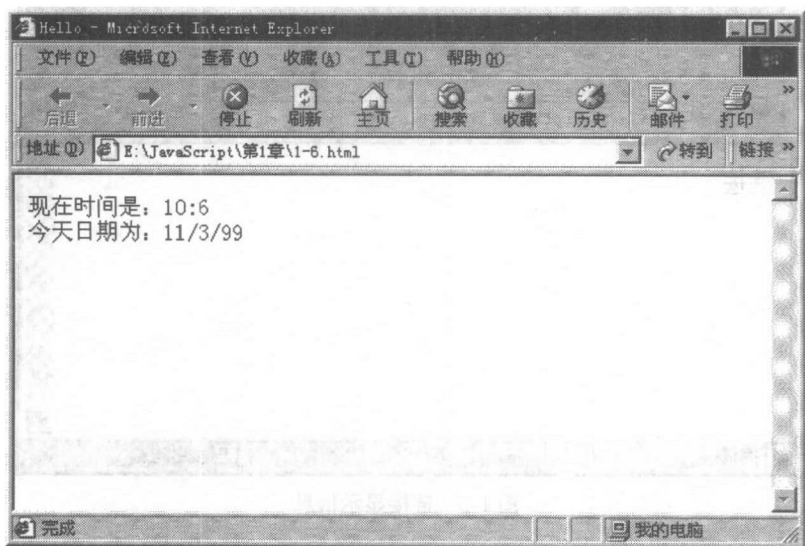


图 1-6 时间和日期显示

在本例中，首先建立一个日期变量，这可以由 `today= new Date()`来完成。如果没有特别指定时间和日期，浏览器将会采用本地客户端系统的时间，并将它放入变量 `today` 中。需要注意的是，在 JavaScript 中，并没有定义 `today` 这个变量的类型，而 Java 或其他的语言则有所不同，在用到变量之前必须先对其加以定义。建立了 `today` 这个日期变量后，等于建立了一个具有本地时间与日期的对象 (object)。以后就可以使用 `get` 方法以取得 `today` 这个对象的时间和日期。

注意，`getMonth` 这个方法所取得的月份范围是由 0~11，所以必须加 1 以代表真正的 1 月至 12 月。

除此之外，在建立日期对象时也可以事先设定日期如下：

```
docStarted= new Date (96, 0, 13)
```

首先是年（公元），接着是月（但记得减 1），再接着是日。同样也可以加上时间的设定，如下：

```
docStarted = new Date (96, 0, 13, 10, 50, 0)
```

前 3 个是日期的年、月、日，接着是时、分、秒。最后，我们必须提醒读者，JavaScript 并没有实际的日期类型，但是它却能毫不费力地显示出日期和时间，原因是它是从 1/1/1970 0:0h 开始以 ms（millisecond）来计算当前时间的，这听起来似乎有些复杂，但不用担心，它有标准的共用函数可以计算，只要知道如何用就可以了。

1.2.7 实例七

下面将向读者介绍一个可以产生随机数的函数，这个函数只是利用了一点小技巧，而在大部分的编译器（compiler）中，都是利用这种技巧（或类似）计算出随机数来的。以下是此函数的结果：

这是一个计算产生的随机数：3496448567590426

源代码如下，图 1-7 为程序运行的结果。

```
<html>

<head>

<script language="JavaScript">
function RandomNumber() {
    today = new Date();
    num = Math.abs (Math.sin (today.getTime()));
    return num;
}
</script>

</head>

<body>
<script language="JavaScript">
<!--
    document.write ("这是一个随机数: ", RandomNumber());
// -->
</script>
</body>

</html>
```

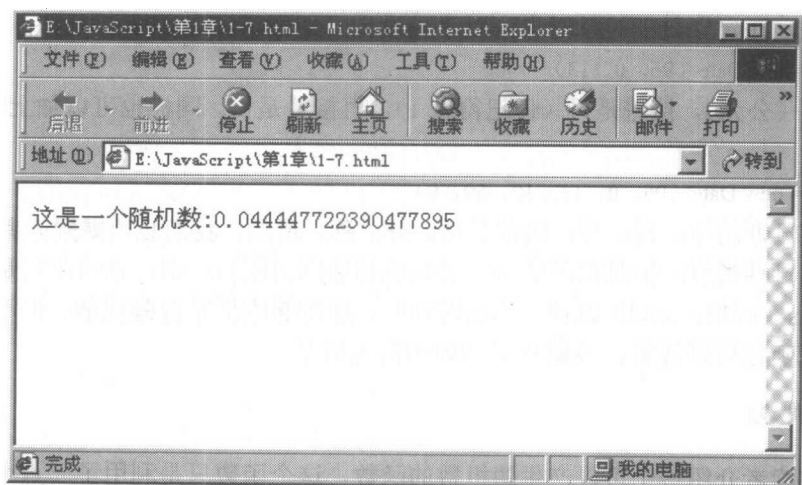


图 1-7 随机数的产生

首先用前面实例中介绍的时间函数产生一个很大的数，再对这个数加以运算。例如：对其做正弦函数（sin）的运算，得到的数再取绝对值，可以得到一个介于 0 与 1 间的实数。因为时间的改变是 ms 为单位，因此理论上不会产生相同的数字。

这个做法并不适合快速、连续地产生一系列的随机数。如果读者只是需要不定时或间隔较久地产生随机数，那效果还是不错的。

1.2.8 实例八

JavaScript 脚本语言的一个重要特点是它可以用来产生一个窗口，并在此窗口中载入 HTML 文件，达到漫游因特网的目的。

下面的实例告诉读者如何产生一个窗口并且写点东西进去，可先试试按一下实例中的按钮。

源代码如下，运行结果如图 1-8 所示。

```
<html>
<head>
<script language="JavaScript">
function WinOpen() {
    msg=open("", "DisplayWindow", "toolbar=no, directories=no, menubar=no");
    msg.document.write("<HEAD><TITLE>哈罗!</TITLE></HEAD>");
    msg.document.write("<CENTER><H1>你好</H1><h2>这是<B>JavaScript</B>所开的窗口!</h2></CENTER>");
}
</script>
</head>
<body>
<form>
```

```
<input type="button" name="Button1" value="启动" onclick="WinOpen()">
</form>
</body>
</html>
```

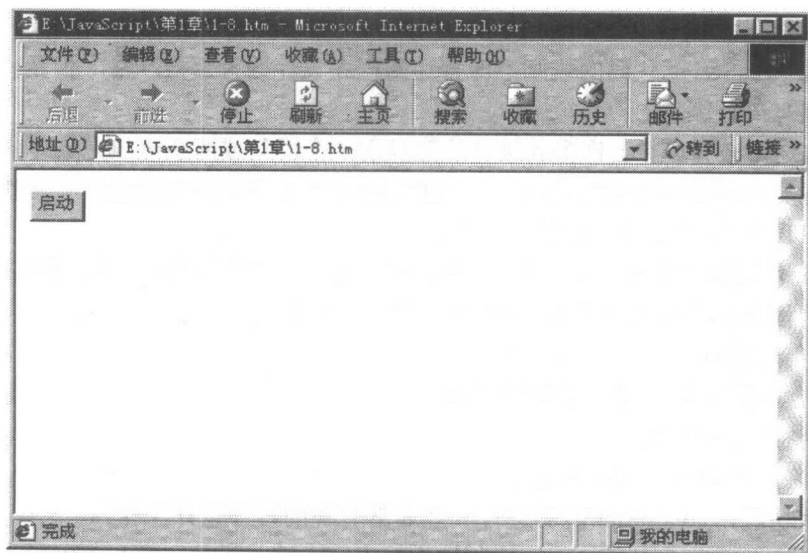


图 1-8 按钮启动函数

这里是利用按钮来启动函数的。函数 WinOpen() 是借 open 方法来产生一个新的窗口。第五行语句中的第一对双引号 (") 是用来描述欲载入窗口中的文件的 URL 位置。如果留着空白的话, 那就会显示一个空白窗口, 读者可以通过 JavaScript 写入东西。下一对双引号的内容是这个新打开窗口的名字, 读者可以任意选择。需要注意的是, 接下来的一连串双引号内所指定的是该窗口的相关性质 (properties), 可以指定要不要有工具条 (toolbar)、滚动条 (scrollbar) 等等, 例如: 如果写 toolbar = yes, 那就会在所产生出来的窗口中出现一排工具栏。以下是一些可以改变的窗口特性, 注意它们中间不可以有空白:

- toolbar
- location
- directories
- status
- menubar
- scrollbars
- resizable
- copyhistory
- width=pixels
- height=pixels

以上的 pixels 部分, 可以填入点数的数值, 这数值是告诉浏览器这个窗口的大小。在打开窗口并且将它设为 msg 以后, 就可以开始写些东西到窗口中了。