

万水 编程实例五十讲丛书

突破

JavaScript

编程实例

五十讲

马健兵 等编著



中国水利水电出版社
www.waterpub.com.cn



清华大学出版社

案例解密

JavaScript

编程实例

五大案例

案例解密



清华大学出版社

万水编程实例五十讲丛书

突破 JavaScript

编程实例五十讲

马健兵 等编著



中国水利水电出版社

www.waterpub.com.cn

内 容 提 要

JavaScript 是一种用于开发 Internet 客户端应用的基于事件的脚本语言,它也是一套与超文本标记语言 HTML 紧密结合的脚本语言,为网页制作者提供了非常灵活的应用和发挥空间。

本书用详尽的实例全面介绍了使用 JavaScript 进行各种客户端编程的过程和应用。全书共分 10 篇,主要包括 JavaScript 的时间日期类、文本、鼠标、图片、页面等特效、页面技巧与页面游戏、导航菜单、Cookie 和 JavaScript 的综合技巧等方面,涉及了 JavaScript 的网络应用的绝大部分内容。针对每个实例给出了实例说明、编程思路、程序说明、实例效果和小结等部分,具有很高的实用价值。

本书内容丰富、实例翔实、文笔简炼,深入浅出、通俗易懂。适用于使用 JavaScript 开发网络应用的用户,尤其适合已有一定 HTML 基础和编程基础的读者。

图书在版编目 (CIP) 数据

突破 JavaScript 编程实例五十讲/马健兵等编著. —北京:中国水利水电出版社, 2002

(万水编程实例五十讲丛书)

ISBN 7-5084-1312-1

I. 突… II. 马… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 094581 号

书 名	突破 JavaScript 编程实例五十讲
作 者	马健兵 等编著
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 68359286 (万水)、63202266 (总机)、68331835 (发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1000 毫米 16 开本 22.25 印张 490 千字
版 次	2003 年 1 月第一版 2003 年 1 月北京第一次印刷
印 数	0001—5000 册
定 价	35.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

《万水编程实例五十讲丛书》

编委会

主 编 万 博 王 龙 韩中领

编 委 苏 瑞 索双有 袁 博 陈海亮 黎 骅 康通博

陈 飞 郭燕婷 关 宁 张美生 文 锋 娄俊杰

杨现青 胡晓冰 商 铮 特 铮 赵庆元 姜仁武

王 静 陈心怡 陈江龙 谢小卓 陈志华 顾卫华

吴志华 王志伟 邹振亚 刘 杨

丛书序

一个优秀程序员和一个普通程序员的区别在什么地方？有一个答案是这样的：普通程序员花了一个星期，苦思冥想如何实现程序功能和排除 Bug。而优秀程序员只用了一天就做完了，因为他给已经做过类似东西的朋友打了个电话。

这段话多半是玩笑，但是也说明了交流的重要。很多读者在学习编程的过程中，早已摆脱了教条式的入门讲解。他们需要的是前人的经验和总结，这就需要交流，看更多的实例，学更多的实用技巧。

如何把前人摸索出来的经验和套路传授给急需它们的读者，如何使计算机的入门教育变成更具有艺术性的教育，是我们一直在寻求的突破口。为此，我们策划并编写了“编程实例五十讲”丛书，从写作到排版风格，从整体结构到每个小例子，我们都做了精心的安排。

在本丛书的编写过程中贯穿了如下的原则：

- (1) 由浅入深、由易到难。整本书结构清晰，适用于初级和中级读者。
- (2) 内容实用、可操作性强。书中对于实例程序行和操作过程的描述要求清楚透彻、简明扼要，编写风格则要求“功能性强”、“示范性强”。
- (3) 知识讲解与动手操作并行。每个实例都突出讲解了某些知识点和使用技巧。知识点与实例的结合是本丛书的一大特点。
- (4) 简单常用的基础实例与综合性较强的应用实例相结合。每本书中都讲解了一些经常会碰到的应用实例，给读者提供思路上的参考。
- (5) 不做简单的功能介绍。本丛书是侧重于应用的图书，我们假定丛书的读者对软件已有一定的了解，或是计算机基础较好，需要快速地在实践中掌握软件的应用。

愿本丛书能带领读者进入程序设计的广阔天地。

编者
2002年4月

我们寻求突破

前 言

JavaScript 是一种解释型的，基于对象和事件驱动的安全性的简单脚本语言。Netscape 在它的浏览器第二版中首先引入了 JavaScript，然后微软（Microsoft）也在它的浏览器 IE 3.0（Internet Explorer version 3）中提供了对 JavaScript 的支持。自此，其他浏览器也陆续开始支持 JavaScript。随着新版本的浏览器不断扩展 HTML，JavaScript 也不断地升级以适应 HTML 的扩展。

JavaScript 脚本语言可以创建和访问 Web 页面中的所有元素，如图片（Images）、表单元素（Form Elements）、链接（Links）等，我们可以在 JavaScript 程序中用合适的方式来读写这些对象的各种属性。JavaScript 还可以捕捉当前的客户端用户对 Web 页的动作，如鼠标的点击或键盘的输入等，这样就使我们能够对用户的动作做出响应，从而实现网页的动态交互。当然，由于它只是一种嵌入式语言，因此它将被嵌入或调入（以 .js 文件的形式）在 HTML 中以语句形式出现。

本书通过具体的实例，详细而深入浅出地介绍了使用 JavaScript 语言进行编程的基本方法和技巧，内容比较丰富和全面。全书共分 10 篇，包括 50 个实例程序，从不同的方面介绍了 JavaScript 的编程方法和应用。

第一篇介绍在使用 JavaScript 的时间日期类进行编程时的各种方法和技巧。第二篇介绍 JavaScript 的文本特效编程的方法和应用。第三篇介绍鼠标特效编程的方法和应用。第四篇介绍图片特效编程的方法和应用。第五篇介绍 JavaScript 的页面特效编程的方法和应用。第六篇介绍 JavaScript 的页面技巧方面的方法。第七篇介绍几个有趣的网上小游戏的制作方法和技巧。第八篇介绍几种网上常见而且耐看的导航菜单的制作方法和技巧。第九篇介绍 JavaScript 在读写 Cookie 以及利用 Cookie 进行一些操作的方法和应用。第十篇综合篇，介绍 JavaScript 在其他方面的一些应用。

本书的开发实例的所有程序代码都在 IE 6.0 环境下编译通过，都经过了严格的调试和测试，读者只要按照书中的步骤，最终一定能够圆满地完成程序。书中的程序源代码可以在中国水利水电出版社的网站（www.waterpub.com.cn）上下载。

本书编写的过程中，得到许多领导和同仁的大力支持和帮助，在此深表谢意！

由于时间仓促以及作者水平有限，书中错误和疏漏之处在所难免，敬请广大读者批评指正。

编者

2002 年 8 月

目 录

丛书序
前言

第一篇 时间日期篇

第一讲	指定位置的时钟显示	2
第二讲	表针式时钟	5
第三讲	带按钮开关的 Form 时钟	11
第四讲	年龄提示器	15
第五讲	得到本文件的最后修改时间	22

第二篇 文本特效篇

第六讲	文字的跑马灯效果	26
第七讲	有提示的链接	34
第八讲	火焰文字	38
第九讲	彩色文字的顺序闪烁效果	41
第十讲	文字从背景中飞出	45
第十一讲	光晕效果的文字	50

第三篇 鼠标特效篇

第十二讲	跟随鼠标移动的彩色星星	56
第十三讲	水中鼠标特效	61
第十四讲	飞船攻击鼠标效果	66
第十五讲	与鼠标联动的动画时钟	78
第十六讲	鼠标星云	87
第十七讲	鼠标焰火	93
第十八讲	Windows 模拟鼠标轨迹	100

第四篇 图片特效篇

第十九讲 图片的淡入淡出幻灯效果	106
第二十讲 图片左右晃动	110
第二十一讲 图片的爆破效果	115
第二十二讲 不断扭曲的图片	122

第五篇 页面特效篇

第二十三讲 星光闪烁的背景	126
第二十四讲 下雨效果	130

第六篇 页面技巧篇

第二十五讲 禁止鼠标右键和键盘的 Menu 键	136
第二十六讲 页面文件的编码和译码	139
第二十七讲 站内搜索引擎	145
第二十八讲 调用其他搜索引擎	155

第七篇 页面游戏篇

第二十九讲 跳舞小游戏	163
第三十讲 测试点击速度	169
第三十一讲 涂格子	174
第三十二讲 数字拼图游戏	181

第八篇 导航菜单篇

第三十三讲 自动滚屏菜单	190
第三十四讲 卷帘菜单的实现	197
第三十五讲 淡入淡出菜单	206
第三十六讲 树状菜单	214
第三十七讲 动态弹出式卷轴菜单	219

第九篇 cookie 篇

第三十八讲 用 cookie 实现日历	226
第三十九讲 统计网友来访次数	235
第四十讲 页面更新跟踪器	239
第四十一讲 自动将访问者的信息 Email 至信箱	242

第十篇 综合篇

第四十二讲	设为首页和加入收藏夹	247
第四十三讲	歌曲播放器	251
第四十四讲	自定义 IE 浏览器中的右键菜单	257
第四十五讲	自定义热键	263
第四十六讲	动态音乐按钮的实现	270
第四十七讲	创建一个科学计算器	273
第四十八讲	仿 OICQ 导航菜单的实现	289
第四十九讲	俄罗斯方块游戏	310
第五十讲	在线测试系统	331

第一篇

时间日期篇

本篇导读

网页中的时间日期的显示和操作是网页的一个重要的组成部分，也是其他很多技巧的基础。因此，JavaScript 提供了十分方便、丰富而又有效的时间日期类的操作，为我们制作网页的该方面的效果作了足够的支持。

本篇对 JavaScript 中处理时间的主要类——Date 类的构造、应用和实现方面作了比较详尽的探讨，同时也讲解了 JavaScript 编程中的一些基本问题，为以后的学习打下了基础。

JavaScript 是一门比较容易入手的语言，读者即使没有编程基础也能在不长的时间内初步掌握它，当然如果是有编程基础的读者，那么掌握这门编程语言要花的时间就更少了。

第一讲 指定位置的时钟显示

【实例说明】

本例所要实现的是在 HTML 页面中的指定位置显示一个时钟。时钟的功能虽然简单，但包含了一个普通 JavaScript 程序的基本要素和特点，因此有助于理解 JavaScript 语言的语法结构和逻辑控制流程。

从实际的代码可以看到，JavaScript 是一种简明易用的基于客户端的操作语言，它可以灵活地嵌入 HTML 页面的不同位置以实现多样的功能。

【编程思路】

编写 JavaScript 仅需要一个文本编辑器就足够了，当然，目前流行的各种 WebBuilder 工具可以更方便地添加 JavaScript 代码，但为了更透彻地学习这种语言，建议使用简单的文本编辑工具。

由于 JavaScript 是嵌在某个特定的 HTML 页面中的，因此首先需要创建 HTML 框架，然后在其中相应位置添加脚本代码。

接下来就需要获得当前系统时间。我们可以用 JavaScript 自带的 Date 对象来获取不同格式的时间。通常可以把具有局部逻辑功能的模块写在一个函数中，JavaScript 中的函数用法和其他编程语言很类似，但更简单，其基本结构为：

```
function functionName(InputArguments) {  
    statements...  
    return(output)    // 非必要  
}
```

若有需要，函数最后可用 return 返回变量（数值、字符串等）。

随后定义了相关的变量来存储时间的各个部分。JavaScript 变量采用的是弱类型，也就是在定义时不必指明其数据类型。一个变量可以是数值（整数和实数）型、字符串型（用“”号或‘’号括起来的字符或数值）、布尔型（使用 True 或 False 表示）和空值，在实际使用时自动指定。

最后需要调用标准输出方法 document.write 在页面上显示出时钟。

【程序说明】

```
<html>  
  <head>  
    <title>指定位置的时钟</title>  
  </head>  
  
  <body bgcolor="#fef4d9" onload=display()>
```

```

<span id=liveclock style=position:absolute;left:100px;top:62px; width:
109px; height: 15px>
</span>
<script language=JavaScript>

```



技巧：函数的定义，通常写在<head>及</head>之间，以确保 HTML 主体在被呈现前，所有相关的 JavaScript 函数都已被载入，并随时可以执行。但这并不是一个严格的规定，通常来说，希望函数的定义出现的位置尽量靠近它被调用的位置，以便程序管理。

```

<!--
function display()
{
    if(!document.layers&&!document.all)
        return
    //document 是 JavaScript 中的一个对象，在它中封装许多有用的方法。由于 IE 与
    //Netscape 对 JavaScript 的解释不同，造成浏览的效果不同，所以要分别写代码。这句
    //话判断一下用户所使用的浏览器，如果 document.layers 为真，则浏览器是 Netscape，
    //如果 document.all 为真，则浏览器是 IE，否则两者都不是，就返回。
    var Digital=new Date()           //新建一个 Date 对象
    var hours=Digital.getHours()     //获得时
    var minutes=Digital.getMinutes() //获得分
    var seconds=Digital.getSeconds() //获得秒
    var dn="AM"                      //一个字符串变量，表示上午或下午

```



说明：JavaScript 变量的命名规则是第一个字符必须为字母或下划线；变量名称中不能有减号或句号；大小写字母视为不同。定义方式分为 Local（局部）和 Global（全局）两种。前者只可使用于副程序或函数，直接设定其值，例如：X=1;Y="字符串"等；后者可使用于程序的任何地方，用 var 定义变量，也可以直接设定数值（多个变量名称以逗号分隔），例如：var X=1;var X,Y,Z 等。

```

    if(hours>12)
    {
        dn="PM"
        hours=hours-12
    }
    if(hours==0)
        hours=12
    //上面两个 if 语句让小时的显示限制在 1~12 之间
    if(minutes<=9)
        minutes="0"+minutes           //“+”号可以用于字符串之间的连接
    if(seconds<=9)
        seconds="0"+seconds
    //改变字体
    myclock="<font size='10' face='Arial' color=#ff00ff><b><font size='5'
color=#66ff00>Current   time   is:</font></br>"+hours+":"+minutes+":"+seconds+"

```

```
" +dn+"</b></font>"
    if(document.layers)
    {
        document.layers.liveclock.document.write(myclock)
        document.layers.liveclock.document.close()
    }
    //document 是 JavaScript 的内置对象，它的 write 方法向页面输出字符串
    else if(document.all)
        liveclock.innerHTML=myclock
    //上面针对不同的浏览器做了不同处理
    setTimeout("display()",1000)        //每隔 1000 毫秒采样一次
    }
    //-->
</script>
</body>
</html>
```



注意：JavaScript 语法的单行注解以 //表示，多行注解以/*...*/表示。而 <SCRIPT>...</SCRIPT>标记间加上<! --... //-->则会让不支持 JavaScript 语法的浏览器将 <SCRIPT>...</SCRIPT>标记间的内容当成 HTML 注解不显示出来。

【实例效果】

在 IE 中打开已经写好的 HTML 页面，可以看到如图 1-1 所示的效果。

Current time is:
12:25:54 AM

图 1-1 简单时钟

【小结】

本讲通过最简单的系统时钟显示介绍了 JavaScript 的一般概念、程序结构特点和部分语法要素。可以看到 JavaScript 语言的文法简单，操作灵活，和 HTML 强大的网页元素描述及表现能力结合在一起更是如虎添翼。JavaScript 本身封装好的各式各样的类能够提供比较复杂的功能。

第二讲 表针式时钟

【实例说明】

通常说来，在页面中套用各种不同格式的文本来显示时间已经足够了，但如果我们想要做得更漂亮些，可以考虑构造一个更为复杂的表针式旋转时钟。

本讲中将用一些点绘成表针状（秒针 5 点，深蓝；分针 4 点，浅蓝；时针 3 点，红色），在数字 1~12 围成的刻度圈上不停旋转。

【编程思路】

实现这个时钟涉及两方面的基本问题：

- (1) 时间的获取：仍然通过构造 `Date` 对象得到。
- (2) 图形的定位：表盘上 1~12 这些数字通过静态设定来放置，而组成表针的共 12 个点则都要通过三角函数做数学定位。其中运用到的几个三角函数，都已经封装在 JavaScript 的内置对象 `Math` 中，可以方便地调用。

目前在 JavaScript 中有 4 种内置型对象：`Date`、`Math`、`string`（各一个对象，`string` 小写 `s`）及一串和 HTML 文档连接的对象。

`Math` 有一套特性（内部常数）包括 `E`、`LN10`、`LN2`、`PI`、`SQRT1_2`、`SQRT2`，所以有 `Math.PI = 3.14159` 等。`Math` 方法包括 Fortran 原有的带例子的库中的常用部分：

`Math.random()` 返回 0~1 之间的伪随机数

`Math.abs(x)` 返回 `x` 的绝对值

`Math.max(x1,x2)` 返回数值 `x1` 和 `x2` 中的最大值

`Math.cos(x)` 返回弧度为 `x` 的角的余弦值

`Math.round(x)` 按照离 `x` 最近的正整数或负整数取整

除此之外，例中还大量用到了 CSS（Cascading Style Sheet，层叠样式表）中页面元素的 `style` 属性。页面上的每个元素都具有一个 `style` 属性，所有 DHTML（Dynamic HTML，动态 HTML）功能都是通过动态地修改 `Style` 属性来实现的。格式如下：

```
style="属性名 1:属性值 1; 属性名 2:属性值 2;...;"
```

上述语句中属性名主要有 `position`、`top`、`left`、`width`、`height`、`background-color`、`color`、`filter`、`font-size` 等。

其中较复杂的是 `Filter`，它有十多种滤镜和二十多种转换。

【程序说明】

```
<html>
```

```

<head>
  <title>表针式时钟</title>
  <script language=JavaScript>
  <!--
    pX=100
    pY=70 //pX 和 pY 是表盘中心的绝对坐标
    pointer=new Array(13)

```



说明：数组定义的方法为 `myarray=new Array()` 或 `myarray=new Array(maxnum)`。

JavaScript 的数组实现是非常灵活的。如果在数组定义的时候没有指定它的大小，那么根据程序的需要它可以自动地伸缩。可以在数组中储存数字、字符或其他对象。此外，同一个数组中的元素也不必是同一个数据类型的，甚至可以让数组的一个元素又是一个数组。例如：

```
myarray=new Array(1,2,3,new Array("a","b"));
```

则：`myarray(0)=1`

`myarray(1)=2`

`myarray(2)=3`

`myarray(3,0)=a`

`myarray(3,0)=b`

数组的 `Length` 属性：array 对象中的 `length` 属性记录了当前数组中最大元素的下标加 1 的数字。

//下面的函数定义了一个数组，它的每一项代表用来组成表盘指针的一个点（时针 3 个点+分针 4 个点+秒针 5 个点），而每项的 3 个域分别存放其 `style`（样式），纵坐标偏移和横坐标偏移（相对于 `pY` 和 `pX`）

```

function pt()
{
  for(i=0;i<13;i++)
  {
    if(document.all)
      pointer[i]=new Array(eval('pt'+i).style,-100,-100)
    else
      pointer[i]=new Array(eval('document.pt'+i),-100,-100)
  }
}

```



技巧：`eval` 是一个特殊的内建函数，包括了运算式子为其参数、运算符、并回传一个值。此函数对于运算一个参数表示的数值字串而言非常有用。

//根据不同浏览器计算终点刻度（1...12）的显示位置

```

function cl(a,b,c)
{
  if(document.all)

```



```
{
  if(a!=0) b+=-1
  eval('c'+a+'.style.pixelTop='+ (pY+(c)))
  eval('c'+a+'.style.pixelLeft='+ (pX+(b)))
}
else
{
  if (a!=0) b+=10
  eval('document.c'+a+'.top='+ (pY+(c)))
  eval('document.c'+a+'.left='+ (pX+(b)))
}
if (document.all)
  c0.style.pixelLeft=26
}
//计算出各指针的绝对坐标
function runClock()
{
  for(i=0; i<13; i++)
  {
    pointer[i][0].left=pointer[i][1]+pX
    pointer[i][0].top=pointer[i][2]+pY
  }
}

var lastsec

//根据不同的浏览器设定组成指针的点的位置
function timer()
{
  time=new Date()
  sec=time.getSeconds()
  if(sec!=lastsec)
  {
    lastsec=sec
    sec=Math.PI*sec/30

    //下面 3 行分别得到秒针、分针、时针的偏转角
    min=Math.PI*time.getMinutes()/30
    hr =Math.PI*((time.getHours()*60)+time.getMinutes())/360
    //下面计算秒针的位置 (由点的 top 和 left 属性设置)
    for(i=1;i<6;i++)
    {
      pointer[i][1]=Math.sin(sec)*(44-(i-1)*11)-16;
      if(document.layers) pointer[i][1]+=10;
```