

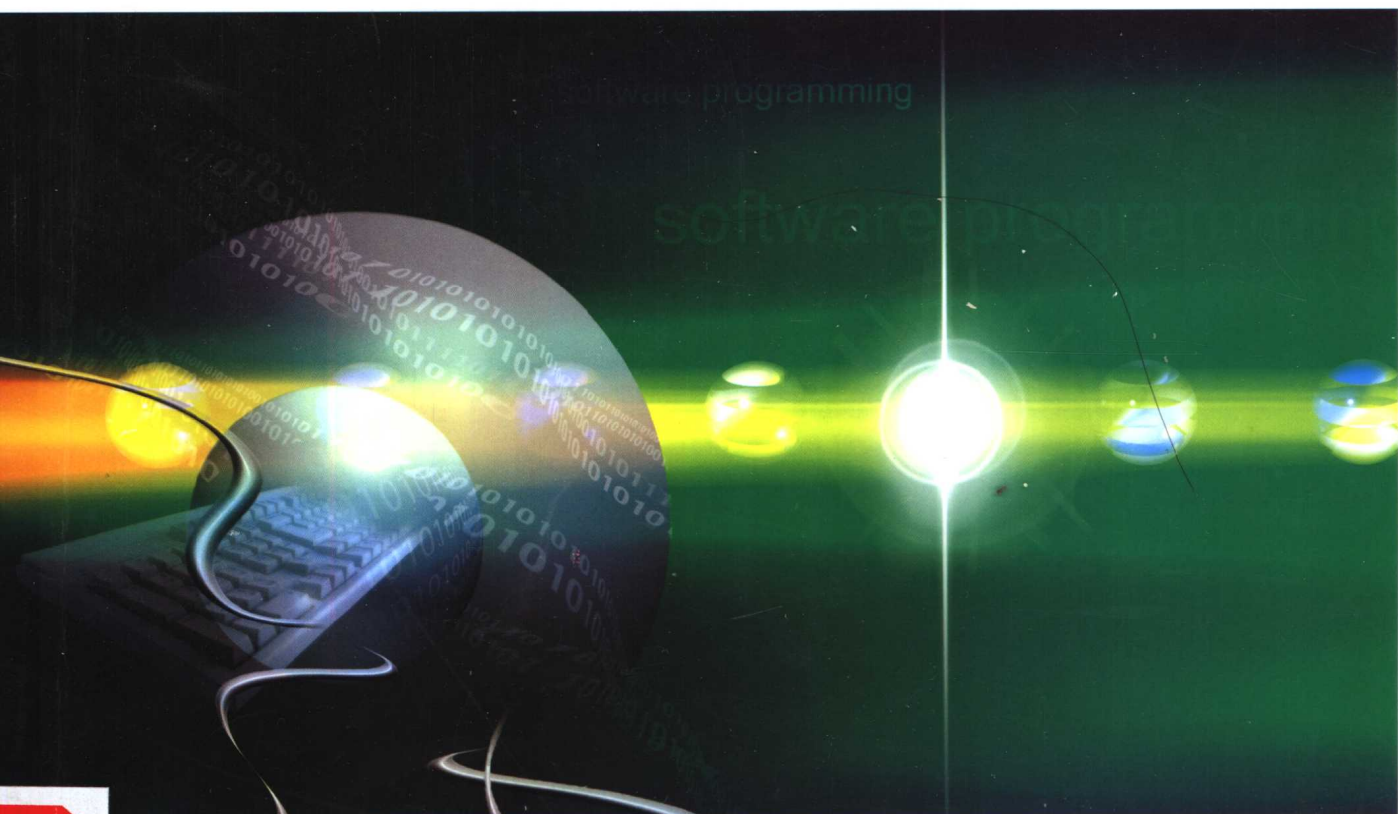
信息科学与技术丛书

程序设计系列

● 李 宇 编著

JavaScript

网页特效实例解析



 机械工业出版社
CHINA MACHINE PRESS



信息科学与技术丛书

JavaScript 网页特效实例解析

李 宇 编著



机械工业出版社

JavaScript 是当今流行的网络脚本语言，本书共对 100 个 JavaScript 例子进行了非常详细的剖析、讲解。本书从 JavaScript 概况开始介绍，然后按特效类型分为 14 章，对各种类型的特效实例进行了详细的讲解，基本上涵盖了 JavaScript 各方面的内容。因为 JavaScript 和 CSS 及 DHTML 联系密切，所以在讲解中也对 CSS 和 DHTML 有关内容进行了一定的讲述。

本书语言简洁、流畅，内容丰富实用，每个例子都可直接使用。学习本书要求读者了解一定的 JavaScript 及 HTML 基础知识（如基本语句及程序流程控制，HTML 基本标签等）。

本书每章对于不同类型的例子，都是由浅入深，内容详实，因此既适合新手学习，也适合具有中等水平的网络程序员及网页设计师学习和参考。

图书在版编目 (CIP) 数据

JavaScript 网页特效实例解析/李宇编著. —北京: 机械工业出版社, 2003.3
(信息科学与技术丛书)

ISBN 7-111-11863-4

I. J... II. 李... III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 019032 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 刘 青

责任印制: 路 琳

北京机工印刷厂印刷·新华书店北京发行所发行

2003 年 4 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·17.25 印张·424 千字

0 001—5 000 册

定价: 31.00 元 (含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

出版说明

随着信息科学与技术的迅速发展，人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书，来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

从所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术受到不断挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机在社会生活中日益普及，随着因特网延伸到人类世界的层层面面，掌握计算机网络技术和理论已成为大众的文化需求。正是这在种社会各领域的全方位渗透，信息科学与技术正在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中得到充分、广泛的应用。相应地，这些专业领域中的研究人员和工程技术人员将越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容，注重理论与实践相结合，内容实用，层次分明，语言流畅，是信息科学与技术领域专业人员学习和参考不可或缺的图书。

现今，信息科学与技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设作出贡献。

机械工业出版社

前 言

随着 Internet 的高速发展, WWW 应用已成为 Internet 中最为重要的部分之一。以前的 HTML 文件只能生成静态的 Web 页面, 不能产生动态效果, JavaScript 脚本语言正是在这种情况下产生的。

利用 JavaScript, 可以在客户端直接对 Web 中的元素进行编程控制, 产生各种动态效果。同时, 因为 JavaScript 是在客户端运行, 所以可以不必依赖于服务端的程序。可以用它设计自己的窗口, 创建自己的对象, 执行数学或统计计算, 建立 Cookie 等。

JavaScript 是一种在客户端执行的解释性语言, 所以不需要一整套的支持工具就可以完成脚本的编辑和调试。只要把含有 JavaScript 脚本的 HTML 文档加载到支持 JavaScript 的浏览器中, JavaScript 代码就可以被解释并执行。

本书对 100 个不同的 JavaScript 例子做了详细的剖析、讲解, 每个例子都具有很强的代表性。本书对每个例子的原理、编程思路及所使用的网页元素属性都进行了分析、讲解, 读者学习之后, 很容易就能达到举一反三、触类旁通的目的。

读者不必掌握太多的计算机专业知识, 只要具有 JavaScript 和 HTML 基础知识, 比如流程控制方法、基本的 HTML 标签等, 就能看懂这本书。本书的例子采用由浅入深、详细讲解的方法, 使读者熟练运用 JavaScript。

本书附送的光盘中含有本书中的所有例子。建议读者在学习过程中多上机操作。只要计算机上装有超文本浏览器和文本编辑器, 就可以直接打开光盘上的例子。通过生动的例子进行学习和运用, 这样一边学习一边上机实践, 可以收到事半功倍的效果。本书中所有例子都在 Internet Explorer 5.0 下测试通过, 大部分例子均可在 Netscape 浏览器下运行。

本书在编写过程中得到了李晓华、黎明、苏磊等的帮助, 在这里表示深深的谢意。如果读者有与 JavaScript 有关的问题, 或是发现本书中的错误, 可通过电子邮件与作者讨论, 地址: future3@yeah.net。

由于时间仓促, 书中难免存在不妥之处, 请读者原谅, 并提出宝贵意见。

编 者

目 录

出版说明	
前言	
第 1 章 JavaScript 网页特效简介	1
1.1 JavaScript 语言简介	1
1.2 JavaScript 和 JScript 的区别	1
1.3 JavaScript 程序设计风格	2
第 2 章 文本特效	5
2.1 在状态栏显示静态文本	5
2.2 在状态栏显示滚动的文字	7
2.3 不断变色的文字	9
2.4 每天显示不同的文本 信息	12
2.5 始终位于窗口顶部的 文本	15
2.6 由小变大的文字	17
2.7 动态水波形文字	21
2.8 动态、闪烁地显示文本的 公告牌	25
第 3 章 窗口特效	29
3.1 自动弹出一个新窗口	29
3.2 自动弹出一个警告窗口	31
3.3 实现与用户的简单交互 功能	33
3.4 自动关闭浏览器窗口的方法	35
3.5 自动变大的窗口	37
3.6 从屏幕顶部下落并抖动的 窗口	39
3.7 在屏幕上左右滚动的 窗口	40
3.8 全屏窗口	43
3.9 无 Windows 风格的窗口	45
3.10 Chromeless Windows—— 另一种风格的窗口	49
第 4 章 时间和日期	59
4.1 在页面上显示日期和 时间	59
4.2 倒记时	62
4.3 在状态栏显示的时钟	63
4.4 定时自动关闭的弹出窗口	64
4.5 用户停留时间	66
4.6 定时提醒用户注意时间	68
第 5 章 链接和按钮特效	70
5.1 变换链接的底色和文本色	70
5.2 几种实用的功能按钮	72
5.3 带浮动条说明的文字链接	75
5.4 会变色的按钮	77
5.5 自动变换文本或背景色的 链接	79
5.6 会动的图形按钮	81
5.7 随鼠标而变的图形按钮	83
5.8 类似 IE 工具栏的按钮栏	85
第 6 章 密码特效	92
6.1 弹出窗口式的密码保护 网页	92
6.2 内嵌在 Web 中的密码保护 网页	94
6.3 具有自加密功能的密码保护 网页	95
6.4 支持多用户的密码保护 网页	99
6.5 检查用户输入密码和用户名 的合法性	103
第 7 章 图片特效	106
7.1 从左向右飞行的图片	106
7.2 随机显示的广告条图片	108
7.3 淡进淡出的图片	110
7.4 在水中倒影的图片	112
7.5 飘动的图片	113
7.6 多幅图片不断轮换显示	116
7.7 多种图片变换的效果	118
7.8 可以用鼠标随意拖动的 图片	120

7.9 飘舞的雪花	124	12.3 测试到某一站点的连接 速度	194
7.10 类似广告切换的效果	127	12.4 获取浏览器的访问记录	196
第 8 章 鼠标特效	132	第 13 章 在线测试	198
8.1 禁止点击鼠标左右键	132	13.1 单项选择测试	198
8.2 点击鼠标右键打开收藏夹	134	13.2 心理分析测试	199
8.3 各种形状的鼠标指针	135	13.3 多题测试评分	202
8.4 在网页上显示鼠标坐标	139	13.4 由颜色来判别你的喜好 个性	206
8.5 用鼠标拖动的网页	140	第 14 章 菜单和导航栏特效	213
8.6 随鼠标移动的瞄准器	142	14.1 简单下拉菜单功能	213
8.7 伴随鼠标左右的跳动文字	144	14.2 下拉菜单	214
8.8 响应鼠标而变化的链接 属性	147	14.3 多级下拉菜单	217
8.9 跟随鼠标移动的文字	149	14.4 浮动在窗口左边的菜单	221
8.10 跟随鼠标移动的图片	152	14.5 右键菜单	225
第 9 章 页面特效	155	14.6 向上滚动的菜单	229
9.1 自动向下滚动的页面	155	14.7 渐显的菜单	231
9.2 页面从中间向两边打开	157	14.8 窗口左边的隐藏菜单	233
9.3 控制用户输入文本个数	161	14.9 可左右滚动的菜单	237
9.4 文本框栏目介绍	163	14.10 一个动态的菜单	240
9.5 会变色的表格框	164	第 15 章 其他特效	244
9.6 变幻的彩色文字	166	15.1 自动显示网页最后更新 时间	244
9.7 旋转的文字	168	15.2 自动选中所有多选项	245
9.8 模仿进度条效果	170	15.3 自动选择速度最快的镜像 站点	247
第 10 章 背景特效	174	15.4 在网页中设置快捷键	248
10.1 背景图片自动定时切换	174	15.5 用 E-mail 方式发送信息	250
10.2 背景颜色选择菜单	176	15.6 多搜索引擎	252
10.3 文本式背景颜色选择器	177	15.7 健康程度检测程序	254
10.4 直观型背景色选择器	179	15.8 计算器	258
第 11 章 Cookie 脚本	182	15.9 判断用户输入的数字是否为 素数	261
11.1 存取 Cookie 信息的例子	182	15.10 对源代码加密	263
11.2 存取用户名的 Cookie 脚本	184	附录 JavaScript 相关资源站点	267
11.3 统计用户来访次数	186	参考文献	268
第 12 章 信息检测	191		
12.1 获得用户浏览器和操作系统 信息	191		
12.2 获得用户屏幕分辨率	192		

第 1 章 JavaScript 网页特效简介

本书对 100 个 JavaScript 例子进行了分析、讲解，这 100 个例子按类型不同共分为十四部分，每部分的例子都由浅入深，并在例子中穿插一些 JavaScript 技术和算法的讲解。灵活使用本书，可以使你在控制并使用 JavaScript 设计制作生动有趣的网页特效方面大大提高。

1.1 JavaScript 语言简介

“JavaScript 是一种易于使用的对象脚本语言，用于创建动态的联机应用程序，这些程序能够将客户端和服务端端的对象和资源连接在一起。Java 是程序设计人员用来创建新对象和小应用程序的，而 JavaScript 是 HTML 页面创作人员和企业的应用程序开发人员用来动态地描述运行在客户端或服务端端的对象行为的。JavaScript 和 Visual Basic 相似的一点在于：即使是很少或者没有编程经验的人也可以用它设计出复杂的应用程序。JavaScript 的设计代表了下一代针对 Internet 的软件设计思想：

- 为创建网络应用程序而设计。
- 补充并集成到 Java 中。
- 与 HTML 集成到一起。
- 开放和交互平台。”

以上这段话是在 1995 年 9 月，Netscape 和 Sun 公司联合发表的声明中对 JavaScript 语言的描述。

JavaScript 是一种可嵌入 HTML 中的，基于对象的解释性脚本语言。它借鉴了许多 Java 语言的语法，但是它和 Java 语言却是完全不同的两种语言。JavaScript 可以方便地控制 Web 中的对象。

它是一种基于对象的语言，所以它并不包括完整的面向对象的编译语言所具备的功能，如 C++。

它是一种解释性语言，所以不需要如编译程序一样准备一整套支持工具（当然，使用工具总比不使用快捷一些）。

1.2 JavaScript 和 JScript 的区别

1995 年 12 月 4 日，Netscape 公司与 SUN 公司合作，推出了 JavaScript 1.0，当时称为 LiveScript。完全不同于以前基于服务器的网络语言，它能在 Netscape Navigator 2.0 浏览器中进行编译操作。作为一种解释性的语言，JavaScript 定位在作为 Java 语言的一种补充（但与 Java 有着本质的区别），它允许开发者通过企业网与因特网来创建和定制应用程序。JavaScript 给予开发者真正编程的能力，而不只是 HTML 的格式数据。

JScript 脚本语言是 Microsoft 公司在其 Internet Explorer 3.0 浏览器中开始实现的解释性

脚本语言。这种技术是 Microsoft 公司在推出 ASP 技术或者动态服务器网页（Active Server Pages）技术前后推出的。

JScript 来源于 Netscape 公司出版的 JavaScript 文档，它应该与 JavaScript 1.0 完全兼容。但是因为 Netscape 公司在其说明书中并没有完全公布 JavaScript 的全部特性，而 Microsoft 公司又没能正确地模拟出来，所以当 Microsoft 发布 Jscript 1.0 时就与 JavaScript 1.0 稍有不同，这也就导致了后来的 JavaScript 和 JScript 必然存在着区别。

而现在 Netscape 公司的 JavaScript 1.5 和 Microsoft 公司的 JScript 5.5 都是依照 ECMAScript 3.0（ECMA 即欧洲计算机制造商联合会）标准而实现的，所以它们之间既有区别又有很多相似之处。

注：ECMAScript 1.0 于 1997 年 6 月正式发行，1998 年 4 月国际标准化组织与国际电子技术协会（ISO/IEC16262）通过了这项标准。

1998 年 6 月，ECMAScript 2.0 被 ECMA 批准。

1999 年 12 月，ECMAScript 3.0 推出，并成为了最新的标准。

ECMAScript 4.0 已经由 Netscape 公司向 ECMA 提出，估计在 2002 年之后的某个时候将成为最新的标准。

在本书中，如果没有特别说明，我们都把 Netscape 公司的 JavaScript 和 Microsoft 公司的 JScript 统称为 JavaScript，毕竟它们还是有很多相同点。

你可以在本书附录中找到更多对以上内容进行详细介绍的站点。

1.3 JavaScript 程序设计风格

不管编程能力的高低，程序设计风格对一个程序员来说都是相当重要的。这一小节将介绍一些在 JavaScript 中的程序设计风格，在后面的例子中都将遵照这些规则，让代码更加易读、易维护、并且易于查找错误。如果你还没有形成很好的程序设计风格，不妨仔细学习一下下面这一段；如果你已形成了很好的编程风格，建议你大体看一下本段，以便能更好地学习本书的例子。

JavaScript 中要注意的程序设计风格主要有以下几点：

- 整体。
- 代码缩进。
- 语句。
- 注释。
- 比较。
- 标识符风格。
- 源文件。

（1）整体

- 要保持整个程序的统一风格，比如要使用代码缩进，那么在平等函数中的语句缩进就要相同。
- 当程序比较大时，最好不要使用全局变量存储数据。好的编程经验就是尽量将某个

对象的所有数据与该对象封装在一起。

- 程序中使用分号风格要一致。因为 JavaScript 中的语句结束可以用分号也可不用，所以在程序设计中，要么都加分号，要么都不加。
- 程序代码的风格就是为了使程序便于阅读和维护。有经验的编程者都知道，其实在程序设计中，代码调试的时间远远多于代码输入的时间。所以保持程序的风格可以很好地增加易读性，便于维护。

(2) 代码缩进

- 代码缩进就是要保持程序层次分明，代码缩进从很大程度上体现出了一个程序员的程序设计风格。
- 在“{”和“}”中设置一级缩进，保证“{”和“}”成对出现并对齐。如：

```
Function test()  
{  
    if (条件)  
    {  
        语句  
    }  
    语句  
}
```

(3) 语句

- 尽量保证语句前后的“{”和“}”成对出现并对齐。
- 如果在设计中要用某一个函数，可是暂时还没写完，最好先定义一个空函数，并说明用处，以便检查。

(4) 注释

- 尽量用 C++ 风格的注释 (//) 代替 C 语言风格的注释 (/* */)，虽然两种注释方法都是允许的。
- 尽量为每个变量和函数加注释。
- 尽量让注释开始处对齐。
- 尽量避免在注释中使用可能已经过时的信息。

(5) 比较

- 在逻辑语句中使用隐含的条件测试；对于其他条件测试（字符串、数字、对象等），使用显式比较。
- 逻辑变量总是一个单独的变量。如：

```
if (逻辑值)  
if (!逻辑值)  
if (数字==0)  
if (浮点变量>0.1)  
if (对象!=NULL) //使用显式比较  
if (对象) //用法错误
```

(6) 标识符风格

- 尽量使用有意义的单词来作为标识符（如英文），这样就可以省去注释。
- 避免出现缩写。
- 除循环控制变量外，不要使用单字符变量名，传统的循环控制变量取名为：i、j、k 等。
- 对于常量，最好习惯使用大写加下划线作为其标识符。
- 对于变量，最好习惯使用小写加下划线作为其标识符。
- 函数名，最好用小写字母开始。
- 对象名，最好以大写字母开始。

(7) 源文件（.js 文件和包括脚本的文件）

- 最好将各个对象放在单独的文件中。
- 每个文件开头应有注释文字。
- 如果一个文件包括多个对象，列出所有对象以及简略描述。

// 文件名： test.js

// 作者： 喜悦

//09/10/2002 建立

第2章 文本特效

文本特效是网页设计中最常用的 JavaScript 特效之一。任何的网页都离不开文本，学习 JavaScript 文本特效是学习 JavaScript 特效制作的基础，学习和使用好 JavaScript 文本特效，对以后的 JavaScript 特效编程学习很有帮助。

2.1 在状态栏显示静态文本

1. 效果

在浏览器的状态栏显示“JavaScript 网页特效实例解析”，如图 2-1 所示。

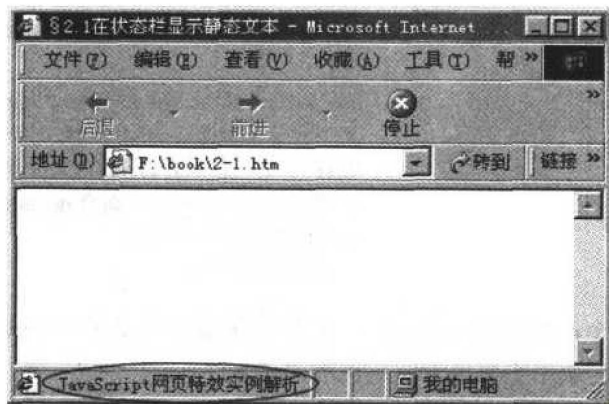


图 2-1 在状态栏显示静态文本

2. 源代码

```
<html>
<head>
<title> § 2.1 在状态栏显示静态文本</title>
<script language="JavaScript">
<!--//Begin code
function SetDefaultStatus(Win,msg)
{
    Win.defaultStatus=msg;
    return true;
}
//End code
-->
</script>
</head>
```

```

<body bgcolor="#FFFFFF">
<script language="JavaScript">
<!--//Begin code
SetDefaultStatus(self.window,' JavaScript 网页特效实例解析');
// -- End code -->
</script>
</html>

```

3. 实例分析

defaultStatus 是 window 对象的一个属性，用它可以控制在当前浏览器的工作窗口底部状态栏中显示的信息。任何时候只要鼠标指针没有经过一个需要显示信息的对象，defaultStatus 属性的值就会显示在状态栏中。如果不给 defaultStatus 属性赋值，那它就会显示一个空串。

程序中，我们在<head>和</head>之间定义了一个函数 SetDefaultStatus()，用它来改变 window 对象的 defaultStatus 属性值。下面对这个函数逐行分析：

```

function SetDefaultStatus(Win,msg){           //定义了函数名及形式参数（简称形参），并开始定义函数内容
Win.defaultStatus=msg;                       //将 window 对象 Win 的 defaultStatus 属性值设为 msg 的值，其中 msg 的值是在调用此函数时传递给形参的
return true;                                  //返回“真”值，告诉 JavaScript 解释程序此事件处理完成；如果为假或无返回值，则给 defaultStatus 赋的值并不生效，即不能改变状态栏的值
}                                              //定义函数结束

```

以上就是定义一个函数时的一般结构。在这个函数中定义了一个改变状态栏的方法，在后面的程序中就可直接调用它来更改状态栏的显示。

在<body>和</body>之间，我们只有一句语句来调用这个函数：

```
SetDefaultStatus(self.window,' JavaScript 网页特效实例解析')
```

它将 self.window 传递给了函数的形参 Win，然后将字符串“JavaScript 网页特效实例解析”传递给了形参 msg，这样，它就通过函数把状态栏的信息改为了我们想要显示的信息。

在本例中也可以不用定义函数，但是使用函数可以让程序更易读，并具有更强的通用性，这是编程的一种好习惯。

技巧：

在程序中，我们把 JavaScript 代码放到了<!--和-->符号之间，这种符号是 HTML 语言中的注释符。当使用的浏览器支持 JavaScript 时，这些符号对 JavaScript 解释执行没有丝毫影响，而当浏览器不支持 JavaScript 时，浏览器会把<!--和-->之间的信息当作注释而不加注意，这样就可以比较好地避免在不支持 JavaScript 的浏览器中运行 JavaScript 时产生出错信息。

虽然现在绝大多数浏览器都支持 JavaScript，但是，让你的程序具有更强的兼容性，而付出的代价只是几个符号，又何乐而不为？

2.2 在状态栏显示滚动的文字

1. 效果

在浏览器的状态栏显示滚动的文字“这是一个状态栏滚动字幕的效果!”，如图 2-2 所示。

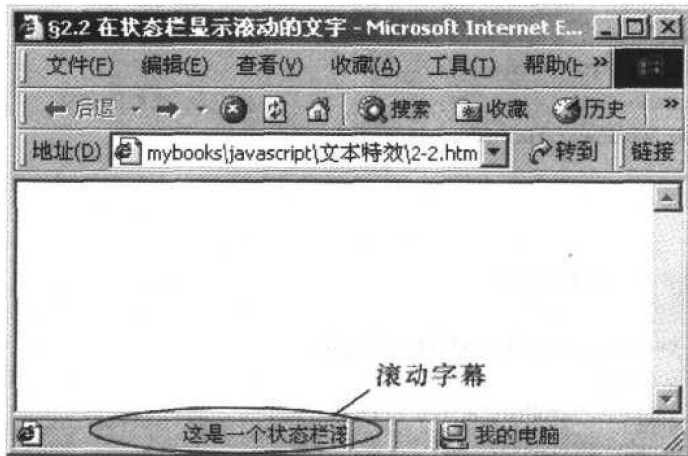


图 2-2 在状态栏显示滚动的文字

2. 源代码

```
<html>
<head>
<title> § 2.2 在状态栏显示滚动的文字</title>
</head>
<body>
<script>
<!-- Begin code --
function scrolltxt(seed)                                //定义显示滚动文字的函数
{
  var str1 = "这是一个状态栏滚动字幕的效果!";
  var str2 = "";
  var msg=str1+str2;
  var out = " ";
  var c = 1;
  var speed= 120;
  if (seed > 100)
  {
    seed=2;
    var cmd="scrolltxt(" + seed + ")";
    timerTwo=window.setTimeout(cmd,speed);            //延迟执行 cmd 中的代码
  }
}
```

```

else
  if (seed <= 100 && seed > 0)
  {
    for (c=0 ; c < seed ; c++)
    {
      out+=" ";
    }
    out+=msg;
    seed-=2;
    var cmd="scrolltxt("+ seed + ")";
    window.status=out; //改变状态栏的文字
    timerTwo=window.setTimeout(cmd,speed);
  }
  else if (seed <= 0)
  {
    if (-seed < msg.length)
    {
      out+=msg.substring(-seed,msg.length);
      seed-=2;
      var cmd="scrolltxt(" + seed + ")";
      window.status=out;
      timerTwo=window.setTimeout(cmd,speed);
    }
    else
    {
      window.status=" ";
      timerTwo=window.setTimeout("scrolltxt(100)",speed);
    }
  }
}
scrolltxt(30);
// -- End code -->
</script>
</body>
</html>

```

3. 实例分析

大家注意到，在本例中我们多次使用到 `setTimeout()` 函数，用它可以在 JavaScript 中很容易地给事件定时，利用 `setTimeout()` 指令可以在未来的某个指定时间执行特定的指令，也就是定时器的作用。如果你改变主意，可以用 `clearTimeout()` 取消 `setTimeout()` 的定时。

其基本格式如下：

```
var the_timeout = setTimeout("some javascript statement",some_number);
```

该函数的功能是在 `some_number` 个毫秒后执行第一个参数 `some javascript statement` (JavaScript 语句)。其中，`some_number` 是一个数字，以毫秒为单位；`some javascript statement`

是一个字符串，可以是一句 JavaScript 语句。the_timeout 是一个变量，它指向特定的 setTimeout。如果想取消该 setTimeout 的定时，只需引用该变量即可。可以给变量起一个别的名称。在本例中有这样的应用语句：

```
var cmd="scrolltxt("+ seed +)";           //将 scrolltxt(seed)字符串赋给变量 cmd,其中在赋值时,
                                          变量 seed 转变成了实际的数字,如:当 seed 等于 28 时,
                                          cmd 中的值为'scrolltxt(28)'
timerTwo=window.setTimeout(cmd,speed);   //在 120ms 后(speed 的值为 120),执行语句 scrolltxt(28),
                                          即 cmd 所包含的值
```

本例编程思想是：定义一个显示文字的 scrolltxt 函数，通过灵活使用 scrolltxt()函数，不断改变 window.status 的值，从而实现了改变状态栏的字幕的目的（window.status 值的内容就是显示在状态栏上的文字，与 defaultStatus 相比，status 属性适用于存放临时的信息）。变量 seed 控制了字幕的动态出现，当它大于 100 时不显示字幕，并将 seed 减 2，再次调用此函数，因为此时显示大于状态栏宽度，看不到效果；当小于 100 大于 0 时，根据 seed 的值，在要显示的内容前加入和 seed 值相同个数的空格，并将 seed 减 2，再通过 setTimeout 调用本函数，每执行一次本函数，seed 的值就比原来小 2，也就是说加在要显示的文字前的空格就少 2 个，这样一来就实现了滚动显示字幕的功能。当 seed<0 时，也就是文字已显示到了状态栏最左边，而当 seed 的绝对值又小于显示文字的长度时，应将欲显示的文字前面相应部分剪去，此功能用 msg.substring(-seed,msg.length)语句实现，它把 msg 从 -seed（这时 seed 本来就是负数）开始到末尾的字符串裁剪下来，再显示。当 seed<0，而其相反数又大于欲显示的文字时，将 seed 的值初始化为 100。这样一直重复执行。

2.3 不断变色的文字

1. 效果

在浏览器中显示一段能不断变换色彩的动态文字，如图 2-3 所示。

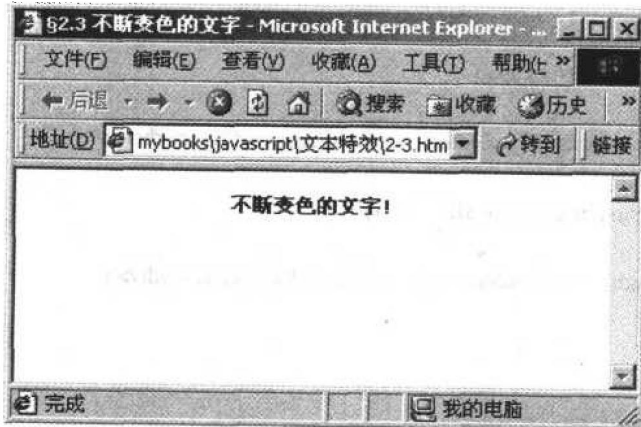


图 2-3 不断变色的文字

2. 源代码

```
<html>
<head>
<title> § 2.3 不断变色的文字</title>
</head>
<body>
<script language= "JavaScript">

<!-- //Begin code --
function initArray()
{
    this.length = initArray.arguments.length;
    for (var i = 0; i < this.length; i++)
    {
        this[i] = initArray.arguments[i];
    }
}
var ctext = "不断变色的文字!";
var speed = 2000;
var x = 0;
var color = new initArray(
"#ffff00",
"#ff0000",
"#ff00ff",
"#0000ff",
"#ffffff",
"#000000",
"#00ff00",
"#00ffff",
"#ff0ff0"
);
if(navigator.appName == "Netscape")
{
    document.write('<layer id="c"><center>'+ctext+'</center></layer><br>');
}
if (navigator.appVersion.indexOf("MSIE") != -1)
{
    document.write('<div id="c"><center><b>'+ctext+'</b></center></div>');
}
function chcolor()
{
    if(navigator.appName == "Netscape")
    {
        document.c.document.write('<center><font color="'+color[x]);
```