

JavaScript

动态网页 经典实例教程

陈争航 编著

姚新军 审校



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

JavaScript动态网页经典实例教程

陈争航 编著

姚新军 审校

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

JavaScript是一种功能强大、语法简单的脚本语言,广泛应用于网页设计中。本书通过具体的实例,详细全面地介绍了使用JavaScript语言结合DHTML进行动态网页设计的方法和技巧。

全书共分14章。第1章和第14章分别介绍JavaScript和DHTML进行动态网页设计的基础知识,并详细给出各个对象的属性、方法的使用技巧,可供网页设计者学习参考。第2章至第13章通过具体的网页设计实例,介绍动态网页设计中的常用方法、重点和难点。实例覆盖现在主流的动态应用——动态状态栏、动态文本框、显示当前时间、动态层、动态内容、动态位置、动态元素内容、滚动效果、动态页面导航、动态数据表单、动态表格等。在动态网页实例的讲解过程中,兼顾主流的浏览器——Internet Explorer和Navigator,使实例的应用面得以拓宽。

本书适合于广大的网页设计从业人员,为动态网页开发设计提供经典应用的解决方案。本书可以作为网页设计爱好者的自学参考书,也可以作为JavaScript动态网页设计的培训教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

JavaScript动态网页经典实例教程/陈争航编著. —北京:电子工业出版社, 2006.3
ISBN 7-121-02257-5

I. J… II. 陈… III. Java语言—网页—制作—教材 IV. TP312

中国版本图书馆CIP数据核字(2006)第006567号

责任编辑:徐云鹏 李红玉

印 刷:北京天竺颖华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:18 字数:450千字

印 次:2006年3月第1次印刷

定 价:26.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换,若书店售缺,请与本社发行部联系。联系电话:010-68279077。质量投诉请发邮件至zlts@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

前 言

JavaScript是一种功能强大、语法简单的脚本语言，广泛应用于网页设计中。使用JavaScript结合DHTML语言，可以实现丰富的动态网页效果。同时，JavaScript也是ASP、PHP、JSP等服务器端语言需要的客户端搭配语言，是进行网页设计所必须掌握的基础脚本语言之一。本书通过大量的网页设计实例，详细全面地介绍了使用JavaScript语言结合DHTML进行动态网页设计的方法和技巧。

全书共分14章。第1章介绍JavaScript语言的基础知识，包括如何在网页中使用JavaScript，JavaScript的基本语法、常用内建对象，并详细给出了各个对象的属性、方法的使用技巧，可作为网页设计者的JavaScript参考手册。第14章介绍DHTML的基础知识，包括常用的对象、属性和方法，可作为DHTML的参考手册。这两章内容帮助读者更好地理解本书的经典实例，同时也方便在实际设计过程中的快速查询。

第2章至第13章通过具体的网页设计实例，介绍动态网页设计中的常用方法、重点和难点。这些实例都是从动态网页主流设计中精选出来的，涵盖各类网页设计中使用的经典动态效果——动态状态栏、动态文本框、显示当前时间、显示当前日期及星期、图像时钟、倒影时钟、网页样式的动态改变、动态修改元素内容、网页内容查找与替换、树形目录、多级树形目录、下拉菜单、带子菜单的下拉菜单、跟随鼠标的图像、跟随鼠标的文字、沿一定路径运动的图片、围绕鼠标的尾巴、位于窗口固定位置的广告、飘落的雪花、彩色雪花、可隐藏的浮动导航面板、可控制公告栏、带停顿的公告栏、滚动的图片串、页面自动跳转、2级级联下拉列表、3级级联下拉列表、禁止使用右键、加入收藏夹、查看源代码、离开时询问、动态数据表单、动态表格等内容。

本书的读者对象

本书是在分析各大主流网站的动态网页应用的基础上，将复杂的应用剖分为合理的应用模块来讲解的。对于广大的网页设计从业人员，通过本书能找到经典应用的解决方案；对于网页设计爱好者，本书可以作为进步的阶梯；对于培训学校，从本书中可以找到实用的JavaScript动态网页设计的培训内容。

本书特色

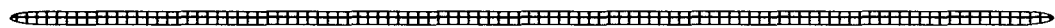
实用性极强——本书所选取的实例，均是在动态网页开发设计中所需要解决的实际问题，均是网站中常用到的效果，都是经典的实例。

快速生产力——读者只需将某些代码更改为自己相应网页中相应的内容，便可以很方便地在自己的网页上实现同样精彩的动态效果。

专注于开发实际——重点介绍程序的设计思想、涉及的关键属性和方法、设计过程中的关键问题，使读者能够举一反三，运用所学知识设计更多实用程序。同时，在设计实例的过程中，不仅考虑了网页的功能，还对网页的美观、布局进行了考虑。

致谢与说明

本书由陈争航老师主笔编著，在编写的过程中，获得电子科技大学边海龙博士的大力支持。此外，张有志、卫喆、程俊、敬宓、王晓艳、万雷也参与部分内容的编写工作。全书的审校工作由姚新军完成。本书在创作过程中得到电子工业出版社及北京美迪亚电子信息有限公司的老师们的的大力帮助、支持和理解。由于时间仓促、任务重、作者水平有限，书中疏漏之处难免，希望广大读者批评指正。



为方便读者阅读，本书配套资料请登录“华信教育资源网”
(<http://www.hxedu.com.cn>)，在“教学资源”频道的“综合资源下载”栏目下载。



目 录

第1章 JavaScript基础	1
1.1 JavaScript简介	1
1.2 JavaScript如何实现网页动态	1
1.3 将JavaScript加入HTML中	3
1.4 解决非JavaScript浏览器问题	5
1.5 加入注释	6
1.6 JavaScript在HTML中的位置	7
1.7 事件与事件驱动	8
1.8 JavaScript语言基础	8
1.9 JavaScript对象模型	23
1.10 JavaScript常用内置对象	25
1.11 全局常量与全局函数	36
1.12 自定义对象	38
1.13 本章回顾	40
第2章 JavaScript内建对象应用	41
2.1 JavaScript计算器	41
2.2 显示当前星期	49
2.3 显示当前时间	51
2.4 应用image对象实现动画	52
2.5 style对象应用	54
2.6 本章回顾	58
第3章 动态状态栏效果	59
3.1 知识要点	59
3.2 文字首尾相接循环滚动	60
3.3 文字由左端弹出	63
3.4 文字由右端飞入	65
3.5 本章回顾	67
第4章 动态文本框效果	68
4.1 知识要点	68
4.2 单行文本框动态效果	68

4.3 多行文本框动态效果	70
4.4 本章回顾	74
第5章 应用时间	75
5.1 知识要点	75
5.2 数字时钟	75
5.3 图像时钟	78
5.4 倒影时钟	81
5.5 日历生成器	85
5.6 本章回顾	92
第6章 动态内容	93
6.1 知识要点	93
6.2 动态修改元素内容	98
6.3 动态添加内容	101
6.4 实现查找与替换	103
6.5 本章回顾	107
第7章 动态层	108
7.1 知识要点	108
7.2 树形目录	117
7.3 下拉菜单	133
7.4 带子菜单的下拉菜单	139
7.5 多级树形目录	146
7.6 本章回顾	152
第8章 动态位置	153
8.1 知识要点	153
8.2 跟随鼠标运动的图片	154
8.3 跟随鼠标的蛇形文字	158
8.4 沿一定路径运动的图片	162
8.5 围绕鼠标旋转的尾巴	164
8.6 位于窗口固定位置的广告	170
8.7 飘落的雪花	174
8.8 隐藏的浮动导航面板	179
8.9 本章回顾	185



第9章 实现滚动	186
9.1 知识要点	186
9.2 可控公告栏	187
9.3 带有停顿的公告栏	191
9.4 滚动的图片串	196
9.5 本章回顾	203
第10章 页面跳转	204
10.1 知识要点	204
10.2 页面跳转	205
10.3 自动跳转	208
10.4 本章回顾	210
第11章 动态数据表单	211
11.1 知识要点	211
11.2 摄影作品欣赏	214
11.3 本章回顾	219
第12章 动态表格	220
12.1 知识要点	220
12.2 客户通信录	221
12.3 可排序图书目录	224
12.4 本章回顾	227
第13章 其他常用效果	228
13.1 禁止鼠标右键	228
13.2 添加到收藏夹	230
13.3 查看源代码	231
13.4 离开时询问	232
13.5 离开时添加到收藏夹	234
13.6 2级级联下拉列表	235
13.7 3级级联下拉列表	239
13.8 调用cookie	245
13.9 本章回顾	250
第14章 浏览器文档对象模型	251
14.1 文档对象模型简介	251
14.2 事件模型	254

14.3	window对象	255
14.4	history对象	260
14.5	location对象	261
14.6	navigator对象	262
14.7	screen对象	263
14.8	frame对象	263
14.9	document对象	263
14.10	form对象	265
14.11	image对象	267
14.12	link对象	268
14.13	anchor对象	269
14.14	元素对象	269
14.15	style对象	270
14.16	cookie	272
14.17	event对象	275
14.18	本章回顾	280



第 1 章

JavaScript基础



1.1 JavaScript简介

在学习JavaScript动态网页设计之前，我们先来了解JavaScript的基本情况，增加对脚本语言的理解。

JavaScript是一种通用的、跨平台的、基于原型的、面向对象的脚本语言。称它为脚本语言，是因为它不同于一般的程序设计语言，它不需要事先进行编译，而是嵌入在HTML文本中，由客户端浏览器对其进行解释执行。JavaScript脚本的作用在于控制网页内容，提供网页浏览者与网页内容之间的交互。同时，JavaScript又不同于用C语言、Perl语言、VB或Java所编写的CGI程序，因为脚本程序是同网页一起发送到浏览器，由浏览器直接进行解释执行的，不需要向服务器端发送请求。

JavaScript语言的前身是LiveScript语言。在Sun公司推出Java语言之后，Netscape公司将Java的程序设计概念融入到原有的LiveScript语言中，并将其更名为JavaScript，也就是最初的JavaScript 1.0版本。JavaScript的正式官方定义是：“JavaScript是一种易于使用的对象描述语言，它是为了创建真正的联机应用程序而设计的，无论是在用户端还是在服务器端，这种应用程序都可以将对象和资源连接在一起。HTML网页设计者和应用程序开发人员均可以使用JavaScript动态地描述在用户端以及服务器端运行的对象的行为”。伴随着Navigator 3.0的推出，JavaScript更新到1.1版本。接着经历1.2、1.3和1.4几个版本，最终到现在最新的1.5版本（在Navigator 6.0版本中运行）。

JavaScript语言编写的脚本程序是作为HTML网页的一部分存在的，是由浏览器直接进行解释执行的，所以JavaScript具有平台无关性。只要客户端的浏览器支持JavaScript，则网页便能正确运行。而现在的主流浏览器Internet Explorer和Navigator均支持JavaScript语言。



1.2 JavaScript如何实现网页动态

JavaScript语言是嵌入到HTML中的脚本语言，它的作用在于控制网页中的对象元素，实现动态交互。

用JavaScript脚本实现网页动态效果，实质上是应用JavaScript自身内建的对象和文档对象模型所提供的对象和事件，对网页内容进行控制，对用户操作进行响应，从而实现网页的动态交互效果。例如，动态改变网页内容，对浏览者鼠标单击、拖动等事件进行响应等。



提示:

文档对象模型 (DOM) 是一种查看文档的方式。网页中的元素通过文档对象模型互相关联, 这样脚本语言可以随时掌握网页上每个对象的情况。关于文档对象模型将在后面的章节中讲述。

下面我们通过一个简单的例子, 来理解JavaScript如何实现动态交互。在下面这个例子中, 在网页放入一按钮, 当用户单击该按钮时, 弹开一个欢迎信息框。

要实现这样的功能, 我们需要具备下面的两个条件:

- 要能够实现弹出对话框的功能。而文档对象模型的window对象提供的alert方法便可以完成该项功能。
- 要判断用户什么时候单击按钮, 这是弹出欢迎信息框的前提条件。系统通过文档对象模型的onclick事件来实现。

提示:

onclick事件是指当用户进行单击时执行的操作。简单地讲就是指当用户用鼠标单击时要执行的代码。关于onclick事件的详细资料, 我们将在下面介绍。

完整的网页代码如下:

例1.1 单击按钮弹出欢迎对话框

```
<html>
<head>
  <title>JavaScript如何实现网页动态</title>
</head>
<script language="javascript">
function welcome( )
{
  window.alert("欢迎光临我们的网站");
}
</script>
<body>
  <p>
    <form>
      单击弹开欢迎对话框     
      <input type="button" value="welcome" onclick="welcome( )">
    </form>
  </p>
</body>
</html>
```

在上面的例子中, 我们定义了一个welcome函数, 在该函数中, 我们应用文档对象模型提供的window对象的alert方法弹出一个对话框 (关于window对象和alert方法, 在以后的章节中会详细进行介绍)。然后, 我们将welcome函数赋给按钮的onclick事件, 这样当用户单击按钮时, 就执行函数welcome中的代码, 弹出一个欢迎对话框。运行效果如图1-1所示。





图1-1 alert方法应用



1.3 将JavaScript加入HTML中

在前面的例子中，我们已经简单地运用了JavaScript脚本语言，下面我们将详细全面地介绍如何将JavaScript脚本加入到HTML代码中。

1.3.1 应用<script>...</script>标记对加入JavaScript

任何脚本语言均可以通过<script>...</script>标记对加入到HTML代码中。要在HTML中加入JavaScript脚本，同样可以使用<script>...</script>标记对。其一般应用格式如下：

```
<html>
.....//html代码
  <script>
    JavaScript代码
  </script>
.....//html代码
</html>
```

添加脚本处

按照上面的格式将<script>...</script>标记对及JavaScript代码加入到网页文档<html>和</html>之间。网页在打开时，浏览器由上至下依次解释文档中的代码，当遇到<script>标记时，<script>标记告诉浏览器脚本代码开始，当接着遇到</script>标记时，通知浏览器本段脚本代码结束。

在使用<script>...</script>标记对加入JavaScript脚本时，一般在<script>标记中加入language属性，告诉浏览器脚本中使用的是何种脚本语言。

格式如下：

```
<script language="javascript">
  JavaScript代码
</script>
```

说明脚本语言的类型

1.3.2 直接加入到HTML标记

当需要使用多句脚本代码时，我们应用<script>...</script>标记对将脚本代码加入到HTML代码中；如果代码比较简单，那么我们可以直接将代码加入到HTML标记中。



在HTML标记中直接加入代码时，JavaScript脚本前必须加上“javascript:”前缀。格式如下：

javascript: 代码

例1.2 标记中直接加入JavaScript

```
<html>
<head>
  <title>JavaScript---弹出欢迎对话框</title>
</head>
<body onload="javascript:window.alert('欢迎光临我们的网站');">
<br>
</body>
</html>
```

直接加入代码的格式

提示：

在这个例子中，我们使用了onload事件，onload是文档对象模型中的一个事件，该事件发生在文档加载前。简单地说就是在文档加载前执行onload事件中的脚本语句。关于Onload事件我们将在后面的章节中详细介绍。

1.3.3 利用SRC属性级联脚本文件

利用<script>标记的SRC属性，可以在网页中使用外部的JavaScript脚本文件。利用SRC属性级联外部脚本文件，可以使得脚本文件重用，提高代码使用效率。

在前面两种加入JavaScript脚本的方法中，当前HTML页中加入的JavaScript代码只能在当前页中使用。当我们需要重复在不同网页中使用相同的脚本代码来完成相同的功能时，我们可以将这些需要共享的代码写入文件，保存为后缀为“js”的文档，然后利用<script>标记的SRC属性来引用。这里要注意的是，后缀名为js的文档文件中，只能包含JavaScript语句，而不应包含其他语句，同样也不要写入<script>标记。

下面是一个使用SRC属性级联外部脚本文件的例子。

例1.3 使用<script>标记的SRC属性级联外部脚本文件

首先，新建一个文档文件，在文件中写入下面的代码：

```
window.alert("欢迎光临我们的网站")
```

将该文件保存为“exa.js”。

提示：

外部脚本文件中只能包含JavaScript语句，而不应包含其他语句或标记。不要将<script>和</script>标记写入该文件中，否则会出错。

然后建立主页页面，在<head>与</head>之间加入如下代码：

```
<script src="exa.js"></script>
```

完整的主页页面代码如下：

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



```

<title>Untitled</title>
<script src="exa.js"></script>
</head>
<body>

```

使用SRC属性级联外部JavaScript脚本文件。

```

</body>
</html>

```

运行结果如图1-2所示。



图1-2 级联js文件



1.4 解决非JavaScript浏览器问题

前面我们已经知道，JavaScript是通过浏览器解释执行的，而与操作系统无关，只要浏览者使用的浏览器支持JavaScript，则JavaScript代码便能在网页中有效执行。但是，如果浏览器不支持JavaScript，则JavaScript代码便会作为文档内容出现在网页中，造成不必要的混乱。

因为我们不知道浏览者使用的是何种浏览器，也就无法了解其使用的浏览器是否支持脚本，所以要防止在遇到不支持JavaScript脚本的浏览器时网页出错。

要解决该问题，可以采取下面两种措施：

- 用<!--与//-->标记。
- 使用<noscript>...</noscript>标记对。

使用<!--与//-->标记时，JavaScript代码块格式如下：

```

<script language="javascript">
<!--
    JavaScript代码
    .....
//-->
</script>

```

当不支持JavaScript的浏览器遇到<!--时，便将<!--与//-->之间的JavaScript代码作为注释文本，而不进行解释执行。而支持JavaScript的浏览器则忽略这两个标记，对JavaScript脚本进行正确的解释执行。



另外，在<!--标记后可以加入任意的注释文本，在//-->标记的//后也可以加入注释文本。

如果你的网页内容全部由JavaScript脚本来实现（实际上这是可能的），那么你就必须使用<noscript>和</noscript>标记对，这样确保在遇到不支持JavaScript脚本的浏览器时对浏览者给以恰当的提示。

应用格式如下所示：

```
<html>
<head>
<title>使用noscript</title>
</head>
<body>
<script language="javascript">
<!--
    JavaScript代码
    .....
//-->
</script>
<noscript>
<br>
<hr>
```

你使用的浏览器不支持JavaScript脚本，所以无法看到本网页的内容。

```
<br>
</noscript>
</body>
</html>
```

在上面的网页代码中，支持JavaScript的浏览器将正确执行<script>与</script>之间的JavaScript脚本，而忽略<noscript>与</noscript>标记之间的内容。而对于不支持JavaScript脚本的浏览器，则将<!--标记与//-->标记之间的JavaScript脚本作为注释内容，而将<noscript>与</noscript>标记对之间的内容作为网页内容。

当然在<noscript>与</noscript>标记对之间我们同样可以使用其他HTML标记来丰富网页内容。



1.5 加入注释

在编写程序的过程中，随时加入注释，有助于增加程序的可读性，不但可以方便其他阅读者，也有助于作者自己日后读懂程序，方便程序日后维护与修改。所以在程序中加入注释是有必要的，在复杂的程序设计中更是有必要的。程序在执行时，注释文本会被忽略，程序中加入注释，除了使程序便于阅读外，并不会影响程序的执行效率。

在JavaScript中加入注释可以采用两种方法。

- 单行注释。
- 块注释。



单行注释采用注释符“//”实现。浏览器遇到“//”符号时，将本行后面的所有内容视为注释内容。该注释符只限于单行注释，当需要加入多行注释时，在每行前面都必须加上该注释符。

单行注释应用格式如下所示：

```
<script>
.....JavaScript代码  //单行注释文本
                        //单行注释文本
</script>
```

提示：

单行注释可以出现在程序中的任何地方。单行注释符“//”后的本行的全部内容都将被视为注释，所以不要将程序代码放在注释符“//”之后。

有时，单行注释并不能满足我们编写程序时的需要。例如我们在定义函数时，往往需要一段注释说明该函数完成的功能，这种情况下可以使用块注释，用/*与*/符号将注释文本块包含起来，这样，位于/*与*/符号之间的所有字符都将被视为注释内容。

块注释应用格式如下：

```
<script>
..... JavaScript代码
/*
  注释文本块
..... 这里的内容均不被浏览器解释
*/
..... JavaScript代码
</script>
```

提示：

需要注意的是，使用块注释不允许嵌套。在/*与*/标记对之间不可以再使用/*与*/标记对。



1.6 JavaScript在HTML中的位置

前面我们了解到，在HTML中加入JavaScript可以使用三种方法：使用<script>...</script>标记对；直接在HTML标记中加入；使用<script>标记的SRC属性级联外部脚本文件。使用这三种方法均可以将JavaScript脚本加入到HTML中。那么，这三种方法应使用在HTML中的什么地方？

实际上，JavaScript脚本可以放在<html>与</html>标记之间的任何地方。如果需要，上面三种方法可以任意多次地使用在HTML代码的<html>与</html>标记之间的任何地方。但是当我们在代码中定义了函数时，就一定要将函数的定义放在对该函数的调用之前。

当浏览器由上至下、由左至右执行HTML代码，遇到对函数的调用时，如果在调用函数之前，该函数已被定义，浏览器将正确执行该函数；如果函数的定义在函数调用之后，浏览器无法执行该函数，将弹出出错对话框。



为了避免发生类似错误，我们一般将函数的定义放在<body>标记之前，这样在调用函数之前，确保函数已被定义。



1.7 事件与事件驱动

如前所述，JavaScript是通过自身内建的对象和文档对象模型所提供的对象和事件，对网页内容进行控制、对用户操作进行响应，从而实现网页的动态交互效果的。在HTML中，可以利用JavaScript直接在网页中添加内容，而在实际中应用更多的是对事件的处理。

所谓事件，简单地说，就是某种状态的变化或用户执行的某种操作。例如，鼠标单击、关闭网页等操作都是事件。事件响应就是对某种状态的变化或操作的发生而进行的响应，对于Web页来说就是对网页浏览用户的操作或某种状态的改变进行的响应。例如，例1.1中用户单击按钮打开欢迎对话框，就是对用户单击进行的响应；例1.2中在网页加载时弹出对话框，是对网页加载进行的响应。这些都是利用对事件的响应来完成的。

在浏览器中，所有事件都由浏览器检测，当浏览器检测到某个事件要发生时，就会自动寻找相应的事件响应函数来响应这个事件。事件响应函数（也叫句柄）就是事件发生时自动执行的函数。一般情况下，事件响应函数的名称是在事件名前面加上on。例如鼠标单击事件的响应函数onclick()。在例1.1中，我们将welcome()函数赋给按钮的onclick()响应函数，这样，welcome()函数便成为鼠标单击按钮的响应函数，当单击操作发生时，浏览器便去寻找该函数，并将控制权暂交给welcome()函数，对用户单击按钮操作进行响应。

浏览器正是通过对事件的响应来实现与网页浏览者之间的动态交互的，而JavaScript脚本的应用就在于编写事件响应函数，来动态响应用户操作，动态改变网页内容。

JavaScript中可以使用许多事件，这些事件根据HTML标记的不同而有所不同，我们将在浏览器对象中介绍。



1.8 JavaScript语言基础

任何语言均有自己完整的一套语言编写规则，JavaScript也不例外。作为Web创作语言，JavaScript既有一般编程语言的共同特点，也有其自身独特的编程风格。

对JavaScript脚本语言有了一个直观的概念后，下面我们来详细学习一下JavaScript语言的基础知识，掌握这些基础知识是使用和学习JavaScript的前提。

1.8.1 数据类型

在JavaScript程序编写过程中可以使用下面的四种数据类型：

- 数值型：JavaScript中的数值型包括整数型和实数型。
- 字符型：字符型是指由单引号或双引号括起来的一串字符。例如：“welcome”，“你好”，“1203-003”在程序中均被视为字符型。
- 逻辑型：逻辑型就是真或假、是或否，这种类型只有两种取值：true或false。
- 空值型：JavaScript中的一种特殊类型，在定义变量时，可以将null值赋给变量。当作

