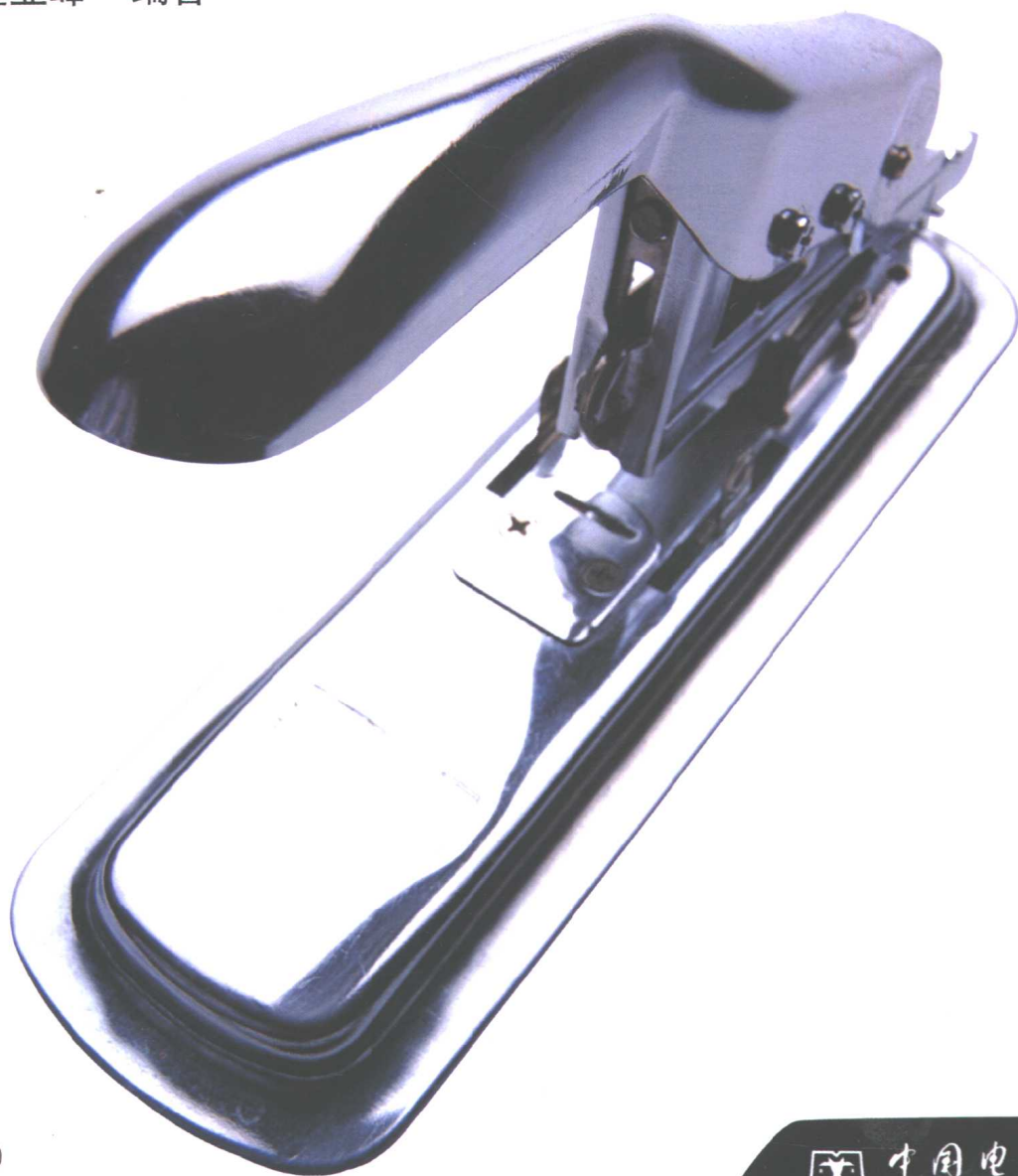


实践与提高丛书

JavaScript

实践与提高

葛蒙 程显峰 编著



中国电力出版社

www.infopower.com.cn

实践与提高丛书



JavaScript 实践与提高

葛蒙 程显峰 编著

中国电力出版社

内 容 提 要

本书以翔实的实例和丰富的内容, 深浅相宜地介绍了 JavaScript 设计语言的编程方法和编程技巧。本书主要包括 JavaScript 语言的特点和功能、JavaScript 的基本语法、JavaScript 在浏览器中的相关实例、JavaScript 的 window 对象、JavaScript 的 document 对象、JavaScript 的日期对象、JavaScript 的 image 对象、JavaScript 的 event 对象以及其他相关对象的应用实例。本书所附光盘含有与该语言相关的动态演示文件, 以方便读者学习使用。

本书通俗易懂、内容丰富, 适合广大计算机程序设计人员、大专院校的师生、计算机网络爱好者和各种网页设计培训班参考使用。

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

北京市地矿印刷厂印刷

各地新华书店经售

*

ISBN 7-900109-03-X/TP·04

2002 年 6 月第一版 2002 年 6 月北京第一次印刷

787 毫米×1092 毫米 16 开本 19 印张 462 千字

定价 33.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题, 我社发行部负责退换)

前言

如果非要给出一个选择 JavaScript 的理由,我想除了我们天性中喜欢生动活泼的本性外,其他理由或许都不重要。如果能够轻松地编写一个简单脚本,却能带来五彩缤纷的精彩变化,我们当然会毫不犹豫地选择它——JavaScript。

JavaScript 是一种基于对象 (Object) 和事件驱动 (Event Driven) 并具有安全性能的脚本语言。我们使用 JavaScript 的最终目的是同 HTML 超文本标记语言、Java 语言 (Java 小程序) 一起实现在一个超文本文档中链接多个对象,并与网络客户实施交互作用。JavaScript 专门用来开发 Internet 客户端和服务端的应用程序,它可以被轻松地嵌入到 HTML 文件中。使用 JavaScript,浏览器可以回应使用者的需求事件而不必通过网络来回传资料。这样,使用者的资料就可以直接被客户端应用程序所处理。它的出现弥补了 HTML 语言的缺陷,是 Java 与 HTML 折衷的选择,该语言具有以下几个基本特点:

- 是一种脚本编写语言

JavaScript 是一种脚本语言,它采用小程序段的方式实现编程。与其他脚本语言类似,JavaScript 同样也是一种解释性语言。它的基本结构形式与 C、C++、VB、Delphi 十分类似。但它不像这些语言一样,需要事先编译,而是在程序运行过程中被逐行解释。它与 HTML 标识结合在一起,从而方便用户的使用。

- 基于对象的语言

JavaScript 是一种基于对象的语言。这意味着它能运用自己已经创建的对象。说它是“基于对象”而非“面向对象”的,这是因为 JavaScript 不支持像 Java 与 C++ 那些语言中那种真正的“面向对象编程”(即 OOP)。但是,JavaScript 确实使用并依赖于对象的,这正是我们称之为“基于对象的语言”的原因。

- 安全性能好

JavaScript 是一种安全性语言,它不允许用户轻易地访问本地硬盘,也不能将数据存入到服务器上,更不允许对网络文档内容进行修改和删除,只能通过浏览器实现信息浏览或动态交互。因此,JavaScript 是一种安全性很高的语言,它能够有效地防止数据的丢失。

- 跨平台性

JavaScript 依赖于浏览器本身,与操作环境无关,只要能运行浏览器的计算机,并支持 JavaScript 的浏览器就可正确执行。比如微软的 Internet Explorer 和网景的 Netscape Navigator,但是在这两个环境中还是有一些细微差别的。

- 交互性和动态性

使用 JavaScript 可以大大增强网页的交互性和动态效果,这也是 JavaScript 最吸引人的功能。读者可以在自己的网页中加入一些可以控制的表单元素,比如通过激发按钮的相应事件来调用相应的脚本语句或者是特定的函数。不仅如此,使用 JavaScript 还可以制作出初学

者难以想像的网页动态特效，这一切都会使你的网页看起来动感十足。

本书采用了理论和实践相结合的方法，一方面详细阐述使用 JavaScript 进行应用程序设计的基本内容，另一方面注重培养读者解决实际问题的能力，书中针对每个主题，提供了一个或几个实例，使读者在学习的同时，能通过自己的实践来加深对该语言的理解，以便在日后的开发中充分发挥 JavaScript 的强大功能。

在随书光盘中提供了书中实例的演示效果和源程序，读者在学习时可随时调用。

希望读者通过对本书的学习，能够熟悉 JavaScript 开发的过程，快速掌握 JavaScript 的开发技巧，并能够自行设计和编写更加生动的网页。

限于时间和作者水平所限，本书不可避免地存在不足和错漏，恳请广大读者批评指正。

作 者
2002 年 4 月

目 录

前 言

第 1 章	JavaScript 简介	1
1.1	JavaScript 的特点	1
1.2	JavaScript 的运行环境	3
1.3	JavaScript 的主要功能	4
1.4	本章小结	4
第 2 章	JavaScript 的基础知识	5
2.1	JavaScript 与 HTML	5
2.2	JavaScript 的关键字	12
2.3	JavaScript 的对象	14
2.4	JavaScript 的数据类型	16
2.5	JavaScript 的常量与变量	16
2.6	JavaScript 的运算符	18
2.7	JavaScript 的基本语句	20
2.8	本章小结	33
第 3 章	实现警告和提示功能	36
3.1	一个最简单的例子	36
3.2	进站时显示提示	39
3.3	单击超级链接显示提示框	41
3.4	移动到超级链接后立刻显示提示	42
3.5	确认链接	43
3.6	离开时显示提示	45
3.7	按照停留时间显示提示	46
3.8	离开时加入收藏夹	49
3.9	本章小结	52
第 4 章	JavaScript 中的状态栏和标题栏	53
4.1	在浏览器的状态栏中显示文字	53
4.2	在浏览器的状态栏中显示动态文字	55

4.3	跟随鼠标变化的状态栏.....	69
4.4	在浏览器的标题栏中显示文字.....	73
4.5	在浏览器的标题栏中显示动态文字.....	76
4.6	本章小结	82
第 5 章	JavaScript 的窗口控制.....	84
5.1	JavaScript 的 window 对象.....	84
5.2	打开窗口	85
5.3	关闭窗口	94
5.4	回到上一个窗口	102
5.5	本章小结	106
第 6 章	JavaScript 的网页背景特效.....	108
6.1	JavaScript 的 document 对象.....	108
6.2	变换背景颜色	109
6.3	变换背景图案	124
6.4	本章小结	128
第 7 章	JavaScript 的时间和日期.....	130
7.1	JavaScript 的日期对象.....	130
7.2	日期的显示	132
7.3	时间的显示	136
7.4	星期的显示	140
7.5	日期对象的综合显示.....	142
7.6	其他常见的特效	147
7.7	本章小结	158
第 8 章	JavaScript 的表单控制.....	160
8.1	JavaScript 的表单对象.....	160
8.2	JavaScript 的文字框对象.....	161
8.3	JavaScript 的按钮对象.....	167
8.4	JavaScript 的单选按钮对象.....	170
8.5	JavaScript 的复选框对象.....	173
8.6	JavaScript 的选择对象.....	176
8.7	JavaScript 的隐藏对象.....	181
8.8	JavaScript 的表单控制实例.....	184
8.9	本章小结	193

第 9 章	JavaScript 的文字特效	194
9.1	样式	194
9.2	CSS 滤镜	196
9.3	使用 HTML 控制文本	198
9.4	JavaScript 中的文本特效	200
9.5	本章小结	228
第 10 章	JavaScript 的图形特效	230
10.1	HTML 的图形引入与控制	230
10.2	JavaScript 的 Image 对象	231
10.3	JavaScript 中的图形特效	237
10.4	本章小结	249
第 11 章	使用 JavaScript 实现鼠标事件类特效	251
11.1	JavaScript 的 event 对象概述	251
11.2	CSS 光标控制	252
11.3	使用 JavaScript 实现鼠标事件类特效	253
11.4	本章小结	267
第 12 章	使用 JavaScript 检测各种信息	269
12.1	JavaScript 中的 navigator 对象	269
12.2	JavaScript 中的 location 对象	272
12.3	JavaScript 中 cookie 的使用	275
12.4	使用 JavaScript 检测各种信息	279
12.5	本章小结	293

第 1 章 JavaScript 简介

本章要点:

- JavaScript 的特点
- JavaScript 的运行环境
- JavaScript 的主要功能
- 本章小结

1.1 JavaScript 的特点

1.1.1 什么是 JavaScript

JavaScript 是一种基于对象 (Object) 和事件驱动 (Event Driven) 并具有安全性能脚本语言, 它所能提供的功能是非常强大的。我们使用 JavaScript 的最终目的是同 HTML 超文本标记语言、Java 脚本语言 (Java 小程序) 一起实现在一个超文本文档中链接多个对象, 并与网络客户实施交互作用。

JavaScript 是一种简洁的面向对象的描述语言, 是专门用来开发 Internet 客户端和服务端的应用程序, 它可以被轻易的嵌入到 HTML 文件中。使用 JavaScript, 浏览器可以回应使用者的需求事件而不用通过网络来回传资料。这样, 使用者的资料就可以直接被客户端应用程序所处理。它的出现弥补了 HTML 语言的缺陷, 它是 Java 与 HTML 折衷的选择, 该语言具有以下几个基本特点:

(1) JavaScript 是一种脚本编写语言。JavaScript 是一种脚本语言, 它采用小程序段的方式实现编程。与其他脚本语言类似, JavaScript 同样也是一种解释性语言, 它提供了一个简单的开发过程。它的基本结构形式与 C、C++、VB、Delphi 十分类似。但它不像这些语言一样, 需要先编译, 而是在程序运行过程中被逐行解释。它与 HTML 标识结合在一起, 从而方便用户的使用操作。

(2) 基于对象的语言。JavaScript 是一种基于对象的语言。这意味着它能运用自己已经创建的对象。说它是“基于对象”而非“面向对象”的, 这是因为 JavaScript 不支持像 Java

与 C++ 那些语言中那种真正的“面向对象编程”（即 OOP）。但是，JavaScript 确实是使用并依赖于对象的，这正是我们称之为“基于对象的语言”的原因。

（3）安全性能好。JavaScript 是一种安全性语言，它不允许用户轻易的访问本地的硬盘，也不能将数据存入到服务器上，更不允许对网络文档内容进行修改和删除，只能通过浏览器实现信息浏览或动态交互。因而，JavaScript 是一种安全性很高的语言，它能够有效地防止数据的丢失。

（4）简单、实用性好。JavaScript 的简单性主要体现在：首先，它是一种基于 Java 基本语句和控制流之上的简单而紧凑的设计，从而对于学习 Java 是一种非常好的过渡。学习过 Java、C 语言或者具有其他编程基础的使用者可以很容易的掌握这门语言。其次，它的变量类型是采用弱类型，并未使用严格的数据类型，这无疑是初学者的一大福音。

（5）动态性能好。JavaScript 是动态的，它可以直接对用户或客户输入做出响应，而无须经过 Web 服务程序。它对用户的反映响应是采用事件驱动的方式进行的。所谓事件驱动，是通过用户在客户端的某些事件来驱动程序。所谓事件，就是指响应用户的动作的程序中的每个用户的具体动作，比如用户按下鼠标、移动窗口、选择菜单等，这些动作都可以视为事件。当事件发生后，可能会引起相应的事件响应。

（6）具有跨平台性。JavaScript 依赖于浏览器本身，而与操作环境无关，只要能运行浏览器的计算机，并支持 JavaScript 的浏览器就可正确执行。JavaScript 在不同的浏览器中的使用方法不尽相同，比如在微软的 Internet Explorer 浏览器与网景的 Netscape Navigator 中，它们的使用是有一些细微的差别的。

实际上，JavaScript 最杰出之处就在于可以用很小的程序做大量的事。无须用户有高性能的电脑，软件仅需一个字处理软件及一个浏览器，无须网络服务器通道，用户通过自己的电脑即可完成所有的事情。

综上所述，JavaScript 是一种新的描述语言，它可以被嵌入到 HTML 的文件之中。JavaScript 语言可以做到回应使用者的需求事件，而不用任何的网路来回传输资料，所以当一位使用者输入一项资料时，它不用经过传给服务器端（Server）处理，再传回来的过程，而直接可以被客户端（Client）的应用程序所处理。

JavaScript 和 Java 很类似，但它们并不是同一种语言。Java 是一种比 JavaScript 复杂许多的程序语言，而 JavaScript 则相对容易了解。写 JavaScript 可以不那么注重程序技巧，许多 Java 的特性在 JavaScript 中并不支持。

1.1.2 JavaScript 和 Java 的区别

虽然 JavaScript 与 Java 有紧密的联系，但却是两个公司开发的不同的两个产品。Java 是 SUN 公司推出的新一代面向对象的程序设计语言，特别适合于 Internet 应用程序开发；而 JavaScript 是 Netscape 公司的产品，其目的是为了扩展 Netscape Navigator 功能而开发的一种可以嵌入 Web 页面中的基于对象和事件驱动的解释性语言。JavaScript 语言的前身是 LiveScript；而 Java 的前身是 Oak 语言。下面，我们对两种语言间的异同作如下比较：

（1）基于对象和面向对象。Java 是一种真正的面向对象的语言，即使是开发简单的程序，也必须设计对象。JavaScript 则是一种脚本语言，它可以用来制作与网络无关的、可与

用户交互作用的复杂软件。它是一种基于对象（Object Based）和事件驱动（Event Driver）的编程语言。因而它本身提供了非常丰富的内部对象供设计人员使用。

（2）解释和编译。两种语言在其浏览器中所执行的方式不一样。Java 的源代码在传递到客户端执行之前，必须经过编译，因而客户端上必须具有相应平台上的仿真器或解释器，它可以通过编译器或解释器实现独立于某个特定的平台编译代码的束缚。JavaScript 是一种解释性编程语言，其源代码在发往客户端执行之前不需经过编译，而是将文本格式的字符代码发送给客户端由浏览器解释执行。

（3）强变量和弱变量。两种语言所采取的变量是不一样的。Java 采用强类型变量检查，即所有变量在编译之前必须作声明。如：

```
Integer a;  
String b;  
a=1111;  
b=2222;
```

其中 a=1234 说明是一个整数，b=4321 说明是一个字符串。而 JavaScript 中变量声明采用其弱类型，即变量在使用前不需作声明，而是解释器在运行时检查其数据类型，如：

```
a=1111;  
b="2222";
```

前者说明 a 为其数值型变量，而后者说明 b 为字符型变量。

（4）代码格式不一样。Java 是一种与 HTML 无关的格式，必须通过像 HTML 中引用外媒体那样进行装载，其代码以字节代码的形式保存在独立的文档中。JavaScript 的代码是一种文本字符格式，可以直接嵌入到 HTML 文档中，并且可动态装载。编写 HTML 文档就像编辑文本文件一样方便。

（5）嵌入方式不一样。在 HTML 文档中，两种编程语言的嵌入方式不同，JavaScript 使用<script></script>标签来嵌入，而 Java 使用<applet></applet>标签来标识。

（6）静态联编和动态联编。Java 采用静态联编，即 Java 的对象引用必须在编译时进行，以使编译器能够实现强类型检查。JavaScript 采用动态联编，即 JavaScript 的对象引用在运行时进行检查。

1.2 JavaScript 的运行环境

JavaScript 的运行环境包括软件和硬件两方面的内容，现分别说明如下：

1. 软件环境

（1）Windows 95/98/ME 或 Windows NT/2000。

（2）Netscape Navigator x.0 或 Internet Explorer x.0。

（3）用于编辑 HTML 文档的字符编辑器（Word、FrontPage、WPS、Notepad、WordPad 等）或 HTML 文档编辑器。

2. 硬件配置

首先必须具备运行 Windows 95/98/ME 或 Windows NT/2000 的基本硬件配置环境。我们推荐的配置为：

- (1) 基本内存 32MB。
- (2) CRT 至少需要 256 颜色，分辨率在 640×480 以上。
- (3) CPU 至少 P_{III}233 以上。
- (4) 鼠标和其他外部设置（根据需要选用）。

1.3 JavaScript 的主要功能

前两节介绍了 JavaScript 的概念、特点和运行环境以及 JavaScript 与 Java 的主要区别，下面介绍使用 JavaScript 可以实现哪些主要功能。

JavaScript 的功能非常强大，在本书后面章节将详细展示 JavaScript 的独特风采，在此，先把 JavaScript 的主要功能概括如下：

1. 交互性

使用 JavaScript 可以大大增强网页的交互性，读者可以在自己的网页中加入一些可以控制的表单元素。比如，用户可以设置一个按钮，并通过激发按钮的相应事件来调用相应的脚本语句或者是特定的函数。不仅如此，用户还可以初步验证即将交给服务器的表单的正确程度。

2. 动态性

使用 JavaScript 可以大大增强网页的动态效果，这也是 JavaScript 最吸引人的一个功能，相信多数的 JavaScript 爱好者和学习者都非常重视 JavaScript 特效的制作。现在可以告诉读者的是，使用 JavaScript 可以制作出初学者难以想象的网页动态特效，但当读者深入学习 JavaScript 之后，就会发现制作这些特效是如此的方便。比如，我们可以制作一些动态的文字或者是控制网页上的图片使其做某种运动等等。这一切都会使你的网页看起来动感十足。

1.4 本章小结

本章主要介绍了 JavaScript 特点、JavaScript 的运行环境和 JavaScript 的主要功能，希望大家通过这一章的学习能够对 JavaScript 有一个初步的了解，为下面的学习奠定一个良好的基础。

第 2 章 JavaScript 的基础知识

本章要点：

- JavaScript 与 HTML
- JavaScript 的关键字
- JavaScript 的对象
- JavaScript 的数据类型
- JavaScript 的常量与变量
- JavaScript 的运算符
- JavaScript 的基本语句
- 本章小结

2.1 JavaScript 与 HTML

2.1.1 什么是 HTML

HTML 被称为“超文本语言”，是编写网页所必须掌握的基本语言之一。而 JavaScript 与 HTML 之间更是密不可分的，所以在这一节中，我们先来介绍一下有关 HTML 的基本常识。

HTML 文件是标准的 ASCII 文件，是包含了许多被称为标签（或称之为“标记”）的特殊字符串的普通文本文件。从结构上讲，HTML 文件是由元素组成的，组成 HTML 文件的元素有许多种，它们被用来组织文件的内容或者编排文件的输出格式。

绝大多数的元素都由元素的起始标签和结束标签组成。元素的起始标签又叫做起始链接签（start tag），元素结束标签也被称为结尾链接签（end tag）。在起始链接签和结尾链接签中间的部分是元素体。每一个元素都有名称和可以供使用者选择的属性，这些属性定义了元素的颜色、外观、名称等信息。元素的名称和属性通常都在起始链接签内被标明。此外，HTML 的标签支持嵌套，也就是说，一个标签可以被放在另一个标签之中。嵌套时，标签不能被穿插嵌套，也就是说，在 HTML 文档中，不能够出现下面的嵌套方式：

```
<html>
<body>
</html>
</body>
```

上述代码正确的嵌套方式应为:

```
<html>
<body>
</body>
</html>
```

从整体上看, HTML 文件就是由一个一个标签嵌套而成的, 这就是 HTML 文件最基本的特点。

下面列出的是 HTML 的常用元素标签, 标签后面是此标签简短的说明。

注意: HTML 文档的标签是不分大小写的。

(1) 基本标记。

- `<HTML></HTML>`: 标记网页的开头与结尾。
- `<TITLE></TITLE>`: 标记网页的标题。
- `<HEAD></HEAD>`: 标记网页头部。
- `<BODY></BODY>`: 标记网页主体。

(2) 结构标记。

- `<BR>`: 强迫断行。
- `<HR>`: 横线。

(3) 显示标记。

- `<EM></EM>`: 斜体显示。
- `<STRONG></STRONG>`: 黑体显示。
- `<CITE></CITE>`: 引用语句, 斜体显示。
- `<CODE></CODE>`: 程式码。
- `<SAMP></SAMP>`: 程式输出样本。
- `<KBD></KBD>`: 键盘输入。
- `<VAR></VAR>`: 程式变数。
- `<BIG></BIG>`: 大号字。
- `<SMALL></SMALL>`: 小号字。
- `<B></B>`: 加粗。
- `<I></I>`: 斜体。
- `<U></U>`: 底线。
- `<STRIKE></STRIKE>`: 删除线。

- `<S></S>`: 删除线。
- `<SUB></SUB>`: 下标字。
- `<SUP></SUP>`: 上标字。
- `<TT></TT>`: 打字机体。
- `<PRE></PRE>`: 依据预定格式, 即用于保持文本正文中的间隔和换行格式。
- `<CENTER></CENTER>`: 对中。
- `<BLINK></BLINK>`: 闪烁。
- `<FONT SIZE></FONT>`: 字体大小。
- `<FONT COLOR></FONT>`: 字体颜色。
- `<FONT FACE=></FONT>`: 指定字形。
- `<SPACER>`: 留白。

(4) 链接标记。

- `<A HREF=""></A>`: 连结。
- `<A NAME=""></A>`: 设定“锚点”。
- `<A HREF=""></A>`: 连结到“锚点”, 若 URL 省略, 则表示在同一网页中。
- `<IMG SRC="">`: 显示图形。
- `<EMBED SRC="">`: 插入 Plug-in。

(5) 表单与表格。

- `<FORM></FORM>`: 表单标签。
- `<SELECT></SELECT>`: 下拉式菜单标签。
- `<TABLE></TABLE>`: 表格标签。
- `<TR></TR>`: 列标签。
- `<TD></TD>`: 行标签。
- `<CAPTION></CAPTION>`: 表格标题。

(6) 框架标记。

- `<FRAMESET></FRAMESET>`: 用于设定浏览器显示图文框的屏幕区域。

(7) 其他标记。

- `<APPLET></APPLET>`: 用来引用 Java Applet。
- `<!-- -->`: 注解标记。
- `<SCRIPT></SCRIPT>`: JavaScript 脚本语言标签。

HTML 文件源码采用如下的基本结构, 这些内容通常是必不可少的。用到的标签均是我们在上面刚刚列出的:

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

```
</head>
<body >
</body>
</html>
```

这种结构在 HTML 文件中通常是固定不变的, 同样, 在我们嵌入脚本语言时, 也不能破坏这种基本结构。

2.1.2 把 JavaScript 嵌入到 HTML 中

在上一节中, 我们介绍了有关 HTML 的常用语法, 虽然介绍的内容非常简单, 但是这些知识为我们下面继续学习 JavaScript 打下了良好的基础, 如果读者想要深入了解关于 HTML 的有关知识, 可以参看相应的专业书籍。在这一节中, 我们将介绍如何把编写好的 JavaScript 语句嵌入到 HTML 文档中。

我们知道, HTML 文档是由若干元素组成的, 每个元素都含有相应的起始标签和结束标签。同样, 作为 HTML 文档的有机部分, 我们要使用相应的标签来完成对 JavaScript 语句的嵌入工作。在编写 HTML 文档的过程中, 我们使用标签<SCRIPT></SCRIPT>把 JavaScript 的相关语句集成到 HTML 的文档中。在每一个 HTML 文件中, 我们可以使用多对<SCRIPT></SCRIPT>标签来嵌入多段 JavaScript 代码, 而每一段 JavaScript 代码中又可以包含一条或多条 JavaScript 语句。

下面, 我们用一个简单的例子来说明脚本是如何被嵌入到 HTML 文档中的:

```
<html>
<head>
<title>
插入脚本
</title>
</head>
<body>
<div align="center">
<br>
    到这里还只是一般的 HTML
<br>

    <script language="JavaScript">
        document.write("这才是 JavaScript 打印出的文字")
    </script>

<br>
    插入结束了
</div>
</body>
</html>
```

正如大家所看到的，代码中的“document.write(“这才是 JavaScript 打印出的文字”)”一段是该例中惟一的 JavaScript 语句。按照要求，我们把它放入到标签<script></script>中，如下所示：

```
<script language="JavaScript">
    document.write("这才是 JavaScript 打印出的文字")
</script>
```

标签<script></script>又被嵌入到标签<body></body>之中。整个代码的效果如图 2-1 所示。

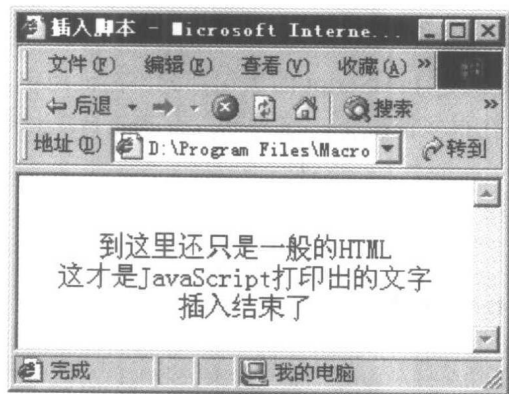


图 2-1 把 JavaScript 嵌入到 HTML 中

在如图 2-1 所示的窗口中，文字“到这里还只是一般的 HTML”和“插入结束了”是由纯粹的 HTML 代码打印出的；而 JavaScript 打印出的是“这才是 JavaScript 打印出的文字”。在今后实例的讲解中，我们将很少单纯的讨论 HTML 代码完成了某种工作，而是把讲解的重点放在如何让大家更好的理解例子中的 JavaScript 代码之上。

其实，脚本语句也可以插入到<head></head>中或<html></html>的其他地方。比如在下面的例子中，我们将把脚本语句插入到 HTML 文档的<head></head>标签中：

```
<html>
<head>
<title>
脚本嵌入
</title>
<script language="JavaScript">
    function pushbutton()
    {
        alert("这是一个简单的例子");
    }
</script>
</head>
<body>
```