

第 1 章 JavaScript 简介

JavaScript 是由 Netscape(网景)公司研发出来的一种脚本程序语言,一般写在 HTML 文件中,能实现网页内容的交互显示。当在客户端显示含有 JavaScript 的网页时,浏览器就会执行 JavaScript 程序,从而用户可以通过交互式操作来变换网页的内容,实现 HTML 语言所不能实现的效果。

开发 JavaScript 的最初动机是希望能在网页上显示时间、动态广告及处理表单、传送数据等工作,以减轻服务器数据处理的负荷。随着 JavaScript 支持的功能日益增多,不少网页制作人员利用它来进行动态网页设计。

1.1 JavaScript 概述

从结构来说,JavaScript 是一种基于对象和事件驱动的安全脚本语言,使用它的目的是与 HTML(超文本标记语言)和 Java 语言一起实现在一个 Web 页面中链接多个对象,从而可以开发出具有丰富表现效果的客户端应用程序。

1.1.1 JavaScript 的历史与标准

JavaScript 由 Netscape 公司的 Brendan Eich 发明,最初叫 Mocha、LiveScript。1995 年,Java 出现后,Netscape 公司引入了 Java 的动态性等设计理念,对 LiveScript 进行了改写,还增加了对 Java Applet 的支持,并因此将其改名为 JavaScript。

1995 年 9 月 18 日,JavaScript 1.0 随 Navigator 2.0 正式版推出。Netscape 公司与 Mozilla 公司后来又陆续推出了 JavaScript 的 1.1 至 1.8 版本。

1996 年 8 月发布的 Netscape Navigator 3.0 支持 JavaScript 1.1。JavaScript 被 Netscape 公司提交给 ECMA(欧洲计算机厂商协会)制定为标准,称之为 ECMAScript,标准编号为 ECMA-262,即 ECMA-262 第一版。

在 Netscape Navigator 4.0 中,JavaScript 升级为 1.2 版本。Netscape 为 JavaScript 1.2 新增了一些功能,但是这些功能没有包含在 ECMA-262 第一版中。ECMA-262 第一版新增了与平台无关这个特性,而 JavaScript 1.2 却无法达到此要求,如 Date 对象就是平台相关的。

Netscape Navigator 升级到 4.06 版本时,JavaScript 的版本更新为 1.3。JavaScript 1.3 修正了“==”和“!=”运算符的功能,并保持与 ECMA-262 第一版中的描述一致。除此之外,所有 JavaScript 1.2 的功能均被保留下来,同时也能与 ECMA-262 第一版兼容。

JavaScript 1.4 集成在 Netscape Navigator 5.0 中,但由于这个浏览器没有发布,因此,这个版本的 JavaScript 也没有面世。

Netscape Navigator 6.0 中引入了 JavaScript 1.5。JavaScript 1.5 和 ECMA-262 第三版完全兼容。SpiderMonkey(用 C 实现的 JavaScript 引擎)和 Rhino(用 Java 实现的 JavaScript 引擎)对 ECMA-262 进行了很多扩展。

JavaScript 1.6 版本应用在 Firefox 1.5 以及其他基于 Gecko 1.8 的产品中,并在以下 3 个方面进行了加强:增加了对 ECMAScript for XML(E4X)的支持,为 Array 对象增加了 2 个定位方法、5 个迭代方法,以及支持字符串和数组泛型。

JavaScript 1.7 涵盖了 JavaScript 1.6 的所有功能,同时又引入了一些新的特性。该版本在 Firefox 2 中引入,同时也应用在其他以 Gecko 1.8.1 为内核的产品中。

JavaScript 1.8 不仅涵盖了 JavaScript 1.6 和 JavaScript 1.7 的所有功能,还在以下两方面进行了细微调整:对简单的函数增加了一种简写的语法形式,为 Array 对象增加了两个遍历函数。

1.1.2 JavaScript 和 Java 的区别

JavaScript 语言和 Java 语言是相关的,但它们之间的联系并不像想象中的那样紧密。下面介绍二者的区别。

(1)它们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,是为了扩展 Netscape Navigator 的功能而开发的一种可以嵌入 Web 页面的、基于对象和事件驱动的解释性语言。

(2)JavaScript 是基于对象的,而 Java 是面向对象的,即 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 是一种脚本语言,它可以用来制作与网络无关的、与用户交互的复杂软件。JavaScript 是一种基于对象和事件驱动的编程语言,因而它提供了非常丰富的内部对象供设计人员使用。

(3)两种语言在浏览器中执行的方式不同。Java 的源代码在传递到客户端执行之前必须经过编译,它可以通过编译器或解释器独立于某个特定平台编译代码。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要经过编译,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。

(4)两种语言使用变量的方式不同。Java 采用强类型变量检查,即所有变量在编译之前必须声明。JavaScript 中的变量采用弱类型,即变量在使用前不需要声明,而由解释器在运行时检查其数据类型。

(5)代码格式不一样。Java 代码的格式与 HTML 无关,其代码以字节的形式保存在独立的文档中,应用时必须像在 HTML 中引用外部媒体那样进行装载。JavaScript 的代码是一种文本格式,可以直接嵌入 HTML 文档,并且可以动态装载。

(6)嵌入方式不一样。在 HTML 文档中,两种编程语言的标识不同,JavaScript 使用 `<script></script>` 来标识,而 Java 使用 `<applet></applet>` 来标识。

(7)静态绑定和动态绑定。Java 采用静态绑定,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 采用动态绑定,即 JavaScript 的对象引用在运行时进行检查。

1.1.3 JavaScript 的基本特点

JavaScript 语言简单、易学,即使不懂程序设计,也可以轻松地在 HTML 文件中嵌入一些 JavaScript 程序代码,实现动态的网页效果。HTML 文件中的 JavaScript 具有以下基本特点:

(1)实现动态的文件内容。JavaScript 可以直接输出 HTML 标签,并使用程序变量更改

输出内容,建立动态的文件内容或图片。

(2)更改 HTML 标签的样式和属性。JavaScript 可以获取 HTML 标签的属性和 CSS 样式,并可以动态更改其内容。

(3)窗体验证和发送。可以使用 JavaScript 编写程序,建立客户端窗体数据的验证规则,在 HTML 窗体数据发送到服务器前验证用户输入的数据是否合理。

(4)处理网页或 HTML 标签的事件。JavaScript 能够处理 HTML 网页或 HTML 标签的事件,如 HTML 文件加载完成、单击按钮等 HTML 标签事件。

(5)Web 应用程序。JavaScript 是在客户端执行的语言,可以在 C/S(Client/Server)模式的应用程序中建立客户端应用程序,还可以与 ASP、ASP. Net 等编程语言一起建立完整的电子商务应用程序。

JavaScript 具有以下优点:

(1)简单性。JavaScript 是一种脚本语言,它采用小程序段的方式实现编程。像其他脚本语言一样,JavaScript 也是一种解释性语言,它提供了一个简单的开发过程。JavaScript 不像 C、C++ 等语言一样需要先编译,而是在程序运行过程中逐行解释执行。JavaScript 与 HTML 标签结合在一起,方便了用户的使用。

(2)动态性。JavaScript 是动态的,它可以直接对客户端输入做出响应,无须经过 Web 服务程序。JavaScript 对客户端请求的响应是以事件驱动方式进行的。以鼠标事件为例,在事件驱动中,用户可以使用鼠标单击等方式进行操作,程序则根据鼠标指针的位置以及单击的方式进行响应。

(3)跨平台性。JavaScript 依赖于浏览器,与操作系统无关,只要是能支持 JavaScript 的浏览器就可以正确执行 JavaScript 程序。

(4)节省 CGI(Common Gateway Interface,公共网关接口)的交互时间。随着互联网的迅速发展,许多 Web 服务器提供的服务需要与用户进行交互,如确认用户的身份等,这项工作通常由专门编写的相应的接口程序来完成。很显然,通过网络与用户进行交互的过程一方面加大了网络的通信量,另一方面影响了服务器的服务性能。服务器为一个用户运行一个 CGI 时,需要一个进程为它服务,这要占用服务器的资源(如 CPU 服务、内存等)。例如,如果用户填写表单出现错误,交互服务占用的时间就会相应增加。被访问的服务器与用户交互得越多,对服务器性能的影响就越大。JavaScript 是一种基于客户端浏览器的语言,表单填写、验证的交互过程是通过浏览器对载入 HTML 文档的 JavaScript 源代码进行解释执行来完成的,浏览器只需将验证用户输入后的信息提交给服务器,从而大大减少了服务器的开销。

1.1.4 JavaScript 的运行平台

在目前的浏览器中,Netscape 公司的 Navigator 2.0 以上版本的浏览器都有处理 JavaScript 源代码的能力。

微软公司的 IE 浏览器从 Internet Explorer 3.0 开始支持 JavaScript,并把其实现的 JavaScript 规范叫做 JScript。JScript 规范与 Netscape Navigator 浏览器中的 JavaScript 规范在基本功能和语法上一致,但在一些对象实现方面有一定的差别,这需要特别予以注意。

1.1.5 不同浏览器对 JavaScript 的支持

由于 JavaScript 代码是在浏览器中解释执行的,因此,了解不同浏览器对 JavaScript 的

支持情况十分必要。

IE 浏览器对 JavaScript 的支持情况及遵循的 ECMA 标准如表 1-1 所示。

表 1-1 IE 浏览器对 JavaScript 的支持及相应的 ECMA 标准

语言版本	浏览器版本	遵循标准
JScript 1.0	Internet Explorer 3.0	ECMA-262 版本 1
JScript 3.0	Internet Explorer 4.0	ECMA-262 版本 1
JScript 5.0	Internet Explorer 5.0	ECMA-262 版本 1
JScript 5.5	Internet Explorer 5.5	ECMA-262 版本 3
JScript 5.6	Internet Explorer 5.6	ECMA-262 版本 3

Netscape 浏览器对 JavaScript 的支持情况及遵循的 ECMA 标准如表 1-2 所示。

表 1-2 Netscape 浏览器对 JavaScript 的支持及相应的 ECMA 标准

语言版本	浏览器版本	遵循标准
JavaScript 1.0	Netscape 2.0	
JavaScript 1.1	Netscape 3.0	ECMA-262 版本 1
JavaScript 1.2	Netscape 4.0~4.05	ECMA-262 版本 1
JavaScript 1.3	Netscape 4.06~4.7	ECMA-262 版本 1
JavaScript 1.4	无	ECMA-262 版本 1
JavaScript 1.5	Netscape 6.x 与 7.x	ECMA-262 版本 3
JavaScript 2.0(建议)		ECMA-262 版本 4(制定中)

JavaScript 与 JScript 版本对照如表 1-3 所示。

表 1-3 JavaScript 与 JScript 版本对照

JavaScript 语言版本	JScript 语言版本
JavaScript 1.0	JScript 1.0
JavaScript 1.3	JScript 3.0
JavaScript 1.5	JScript 5.5

说明:本书以 ECMA-262 版本 3 为基础,并结合 JavaScript 1.5 和 JScript 5.5 来进行相关知识的讲解。

1.2 JavaScript 程序编辑工具

JavaScript 程序代码可直接嵌入 HTML 文件,或独立保存为扩展名为 .js 的程序代码文件,它是一个标准的 ASCII 码文本文件,可以使用文本编辑工具、程序代码编辑工具或 HTML 网页制作工具进行编辑。

1.2.1 纯文本编辑器

可以在纯文本编辑器(如 Windows 记事本)中编辑 JavaScript 程序代码。执行“开始”→

“程序”→“附件”→“记事本”命令,启动记事本程序来编辑 JavaScript 程序代码。编辑好代码后,取消选中“格式”→“自动换行”选项,执行“查看”→“状态栏”命令,在记事本编辑区中单击,即可在其状态栏中显示对应的代码的行号和列号,如图 1-1 所示。



图 1-1 记事本窗口

1.2.2 专业程序编辑工具

专业程序编辑工具很多,这里介绍一种常用的名为 EditPlus 的程序编辑工具。EditPlus 3.0 的 JavaScript 代码编辑窗口如图 1-2 所示。

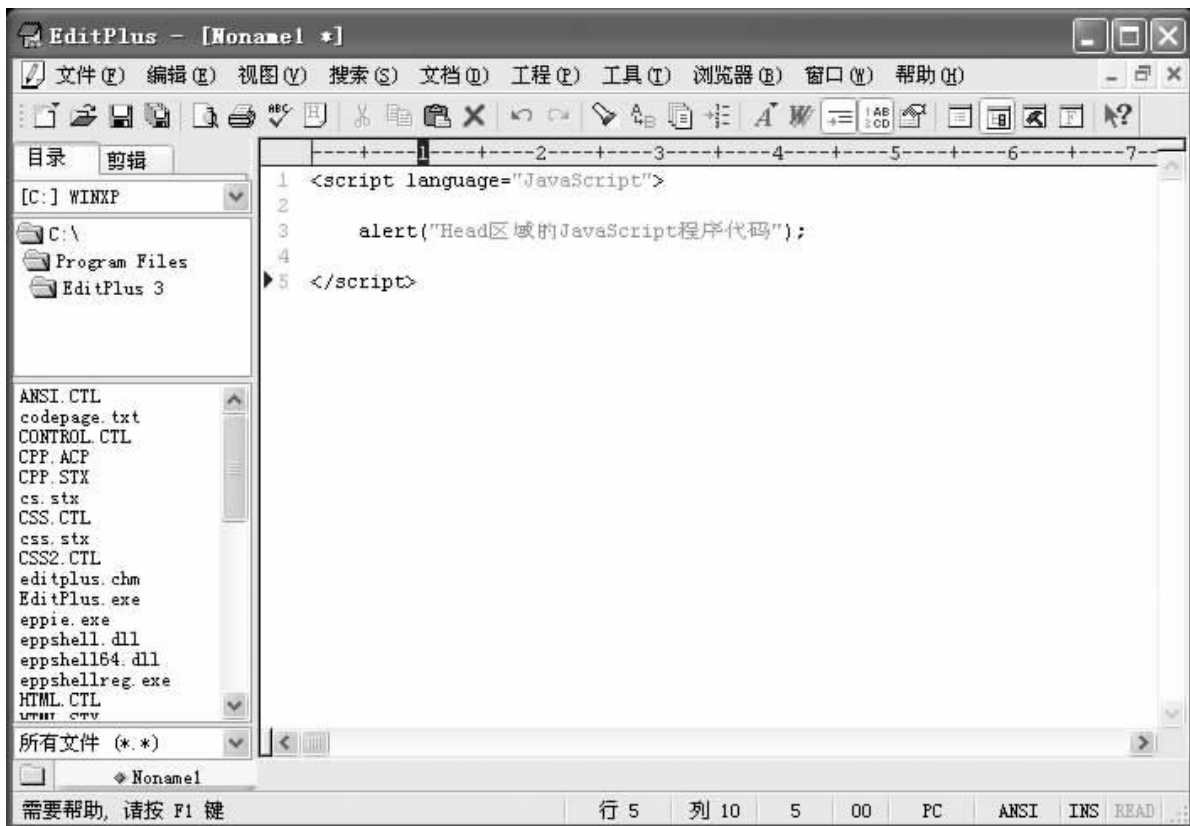


图 1-2 EditPlus 3.0 的 JavaScript 代码编辑窗口

EditPlus 是一个运行在 Windows 操作系统中的文本编辑程序。它不仅可以取代用于文本编辑的“记事本”程序,还可以用作网页及程序代码的编写工具。EditPlus 以鲜明的颜色显示语法,方便用户使用 HTML、PHP、ASP、Perl、C/C++、Java、JavaScript、VBScript 等

语言编写程序。用户可以使用预设的语法颜色,也可以依个人的喜好在如图 1-3 所示的“参数选择”对话框中自定义语法颜色。



图 1-3 设置语法颜色

作为一个专业的程序编辑工具,EditPlus 还具有代码折叠功能,支持多文档编辑,并结合了 IE 浏览器,用户可以直接预览编辑好的网页。

在 EditPlus 中可以直接运行 JavaScript 代码。例如,在 EditPlus 中输入以下代码:

```
<script language="JavaScript">
    alert("Head 区域的 JavaScript 程序代码");
</script>
```

执行“浏览器”→“编辑源文件”命令,运行结果如图 1-4 所示。



图 1-4 运行结果

提示:关于 EditPlus 编辑工具的详细使用方法请读者参考相关资料。

1.2.3 Dreamweaver 网页制作工具

Macromedia Dreamweaver 是建立 Web 站点和 Web 应用程序的专业工具。它将可视化布局、应用程序开发和代码编辑功能组合在一起,功能强大,使各个层次的开发和设计人员都能够利用其快速创建基于标准的、界面美观的网站和 Web 应用程序。

下面简单介绍使用 Dreamweaver 编辑 JavaScript 的方法。

(1)启动 Macromedia Dreamweaver 8,启动界面如图 1-5 所示。

(2)在“创建新项目”选项组中单击“JavaScript”选项,显示 Macromedia Dreamweaver 8 的 JavaScript 代码编辑界面,如图 1-6 所示。



图 1-5 Dreamweaver 8 的启动界面

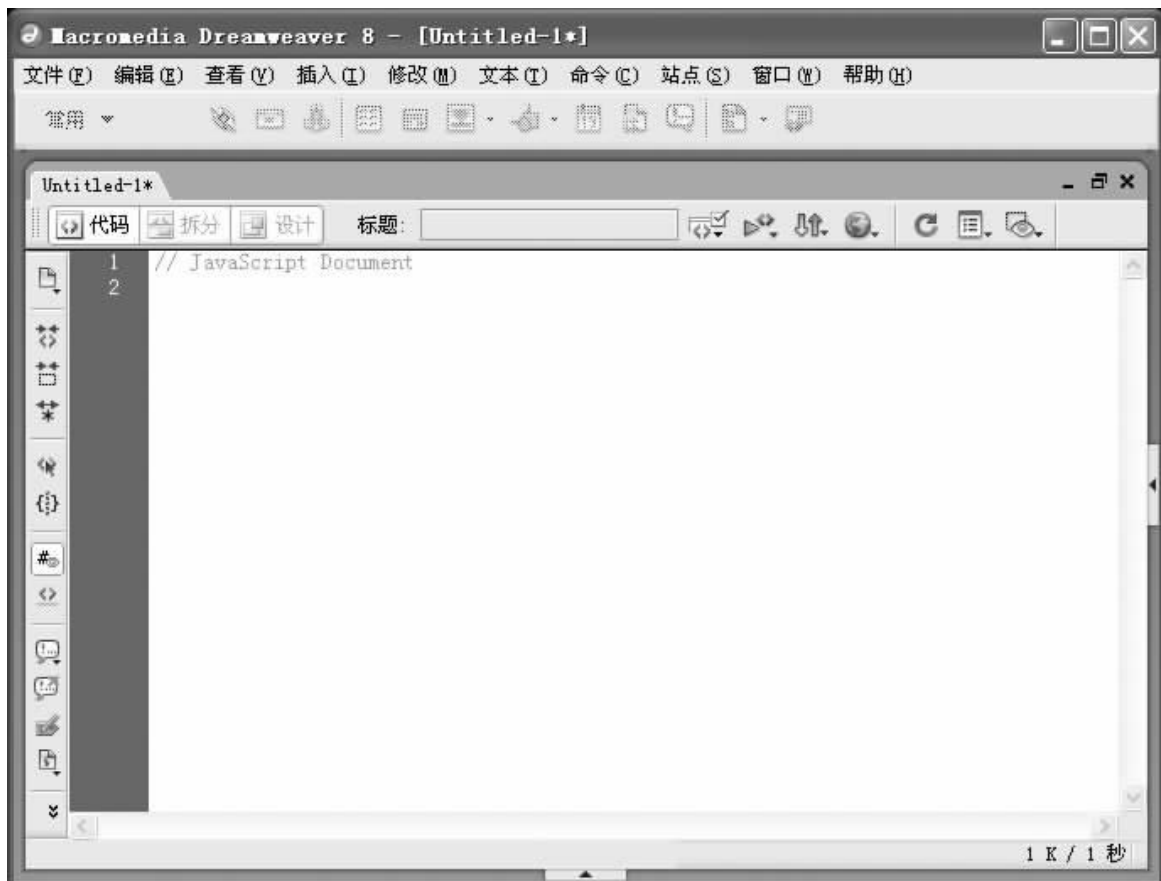


图 1-6 Dreamweaver 8 的 JavaScript 代码编辑界面

(3)在代码编辑区中输入以下 JavaScript 代码:

```
<script language="JavaScript">  
    alert("JavaScript 程序代码编辑区");  
</script>
```

(4)执行“文件”→“保存”命令,在弹出的“另存为”对话框中设置保存路径、文件名和文件类型(“HTML 文档”类型)。

(5)按 F12 键,即可在 IE 浏览器中预览网页效果。

提示:读者可以查阅相关资料对 Dreamweaver 进行深入学习。

1.3 JavaScript 应用简介

JavaScript 是一种功能强大的脚本语言,用于开发交互式的 Web 页面。它不仅可以直接应用于客户端的 HTML 文档,以获得交互效果或其他动态效果,还可以运行于服务器端。

1.3.1 客户端应用

最典型的 JavaScript 应用就是使用其开发客户端 Web 应用程序,即客户端脚本程序。与高级语言不同,客户端脚本程序是解释执行的,即在执行 JavaScript 脚本之前,无须进行编译等预处理。JavaScript 脚本程序被嵌入 HTML 文件,并随 HTML 文件一同下载到客户端的浏览器。浏览器读取 HTML 文件,然后解释执行并显示其中的元素。读取 HTML 文件并分辨其中的元素的过程称为语法分析或解析。如果解析到 JavaScript 脚本,则浏览器执行其脚本语句。

1.3.2 服务器端应用

JavaScript 脚本不但可以运行于客户端,而且可以运行于服务器端,用于实现服务器端的某些特定功能。在运行 Windows 操作系统的服务器上,一种典型的应用就是使用 ASP (Active Server Pages)技术开发应用程序。可以将 JavaScript 作为 ASP 的实现脚本来开发服务器端的应用程序。

服务器端脚本的工作过程为:用户使用浏览器请求 URL 时,服务器调用脚本并对脚本进行处理(对于数据库应用,则需要进行特定的数据库操作),然后将数据以 HTML 文件的形式通过服务器返回给发出请求的浏览器。

1.4 在 Web 页面中使用 JavaScript

JavaScript 程序是嵌入到 HTML 文档中的,因此,任何能编写 HTML 文档的软件都可以用来编写 JavaScript 程序。在 HTML 文件中用<script></script>标签对插入 JavaScript 程序代码,当浏览器读到<script>标签时,就知道内含 JavaScript 程序代码,需要解释并运行这些代码。

1.4.1 HTML 的基本结构

HTML(Hyper Text Marked Language,超文本标记语言)是一种用来制作超文本文档的简单标记语言,它是构成网页文档的主要语言,也是目前网络上应用最为广泛的语言。HTML语法非常简单,它采用简洁的标签命令,通过对文字、图像、多媒体等的定义,创建具有文本、图像、多媒体等对象的复杂页面。

HTML 的语法主要是通过各种标签来实现的。HTML 标签用于指定标签中对象的属性,它是用“<”和“>”括起来的短语和符号,如<html>、<body>等。许多 HTML 标签以成对的方式出现,如<html></html>、<body></body>等。需要注意的是,HTML 语言是不区分大小写的,如<HTML>与<html>的意义是相同的。

1)<html>标签

<html>标签在文档内容的最外层,它表示此文档是以超文本标记语言(HTML)编写的。事实上,现在常用的 Web 浏览器都可以自动识别 HTML 文档,并不要求文档有<html>标签,也不对此标签进行任何操作,但为了使编写的 HTML 文档能够适应各种不同的 Web 浏览器,还是应养成不省略此标签的良好习惯。其格式为:

```
<html>HTML 文档的内容</html>
```

2)<head>标签

<head>标签是 HTML 文档的头部标签。在浏览器窗口中,头部标签中的信息不会显示。可以在此标签中插入其他标签,用以说明文件的标题和一些相关属性。如果不需要头部信息,则可以省略此标签,但最好不省略。其格式为:

```
<head>头部的内容</head>
```

3)<title>标签

<title>标签嵌套在头部标签中,此标签中的文本是文档标题,并在浏览器窗口的标题栏中显示。其格式为:

```
<title>标题名</title>
```

4)<body>标签

<body>标签是 HTML 文档的主体标签,它定义了网页上显示的内容与显示格式,是整个网页的核心。网页中的大部分内容都包含在此标签中,一般不省略。其格式为:

```
<body>文件主体</body>
```

下面这段代码显示了 HTML 文档的基本结构。

```
<html>
<head>
<title>网页的标题</title>
</head>
<body>
网页的内容
</body>
</html>
```

将这段代码保存为扩展名为.html 的文本文件,即网页,并在 IE 浏览器中将其打开,效果如图 1-7 所示。



图 1-7 基本网页

1.4.2 在 HTML 中嵌入 JavaScript 程序

使用`<script></script>`可以在 HTML 文档的任意位置插入 JavaScript 程序。其格式为：

```
<script>
<!--
    (JavaScript 代码)
    // 注释语句
    // -->
</script>
```

其中,第 2 行和第 5 行的符号的作用是让不能识别`<script>`标签的浏览器忽略 JavaScript 代码。由于现在常用的浏览器都能识别此标签,因此,一般可以省略这两个符号。第 4 行中的“//”是 JavaScript 的注释符号,在后面的章节中将对其进行详细介绍。

需要说明的是,如果要在使用框架的网页(框架网页)中插入此标签,应在`<frameset></frameset>`之前插入,否则不会运行。

在一个 HTML 文档中,可以使用多个`<script></script>`标签对,嵌入多段 JavaScript 代码,并且各段 JavaScript 代码之间可以相互访问,如同将所有代码放在一对`<script></script>`标签之中。

1.4.3 链接 JavaScript 文件

如果要在多个 HTML 网页中使用某个 JavaScript 程序,最好的方法是将这个 JavaScript 程序放在一个扩展名为`.js`的文本文件中,而不必将相同的 JavaScript 代码复制到多个 HTML 文档中,这提高了 JavaScript 的复用性。需要修改程序时,只需修改`.js`文件,减少了代码维护的负担。

在 HTML 文档中引用外部 JavaScript 文件时,应在`<head></head>`中加入`<script src="文件名"></script>`,其中,`src`的值就是 JavaScript 文件的路径。例如:

```
<html>
<head>
<script src="http://www.admin5.com/html/asdocs/js_tutorials/common.js">
</script>
</head>
<body>
</body>
```

```
</html>
```

上述代码的效果是链接外部的 JavaScript 文件。其中,语句“src="http://www.admin5.com/html/asdocs/js_tutorials/common.js"”表示外部 JavaScript 文件 common.js 的路径。common.js 文件的内容如下:

```
function clickme()  
{  
  alert("You clicked me!");  
}
```

1.5 编写简单的 JavaScript 程序

浏览器在加载 HTML 文件时,如果读到 JavaScript 程序代码,就会马上执行。理论上可以在 HTML 文件中嵌入无限数量的 JavaScript 程序段。

通常情况下,函数放在 HTML 文件的 head 区域中,即<head></head>内;网页内容或调用函数的程序段放在 body 区域,即<body></body>内。如果程序代码太长,或需要同时使用数个 JavaScript 文件,则可以将 JavaScript 程序代码独立成为一个文件。

浏览器载入网页的<body>标签时,就执行其中的 JavaScript 语句,并将输出的内容显示在网页中,如以下代码所示:

```
<html>  
<head></head>  
<body>  
<script type="text/JavaScript">  
.....  
</script>  
</body>  
</html>
```

【例 1-1】 在<body>标签中编写 JavaScript 程序,用于加载网页内容时显示一个信息窗口。

```
<html>  
<head>  
<title>Body 区域的 JavaScript 程序代码</title>  
</head>  
<body>  
<h2>Body 区域的 JavaScript 程序代码</h2>  
<hr>  
<script language="JavaScript">  
alert("Body 区域的 JavaScript 程序代码");  
</script>  
运行 Body 区域的 JavaScript 程序代码。  
</body>
```

```
</html>
```

其中,第 6 行用于设置网页标题;第 7 行用于添加水平线;第 8 行至第 10 行是嵌入的 JavaScript 程序代码;第 9 行是运行 alert 函数,显示对话框。运行结果如图 1-8 所示。



图 1-8 运行<body>标签中的 JavaScript 程序

弹出如图 1-8 所示的对话框时,网页尚未完全载入,单击“确定”按钮,即可看到网页的内容,如图 1-9 所示。



图 1-9 显示网页内容

有时并不需要一载入网页就运行 JavaScript 程序,而是用户单击了 Web 页面中的某个对象,触发了一个事件后才调用 JavaScript 程序。这时通常将这样的 JavaScript 程序放在 HTML 的<head>标签里,如以下代码所示:

```
<html>
<head>
<script type="text/JavaScript">
.....
</script>
</head>
<body>
</body>
</html>
```

【例 1-2】 在<head>标签中编写函数 showmessage(),触发页面中的 OnClick 事件时调用此函数,弹出一个对话框。

```
<html>
<head>
<title>Head 区域的 JavaScript 程序代码</title>
<script language="JavaScript">
function showmessage(){
alert("Head 区域的 JavaScript 程序代码");
}
</script>
</head>
<body>
<h2>Head 区域的 JavaScript 程序代码</h2>
<hr>
<div OnClick="showmessage()">
在 Head 区域运行事件处理程序。</div>
</body>
</html>
```

其中,第 5 行至第 7 行是 showmessage 函数,第 6 行使用 alert 函数显示对话框,第 11 行至第 12 行是 HTML 文件的内容,用于显示标题文字和一条水平线,第 13 行为 OnClick 事件。运行结果如图 1-10 所示。

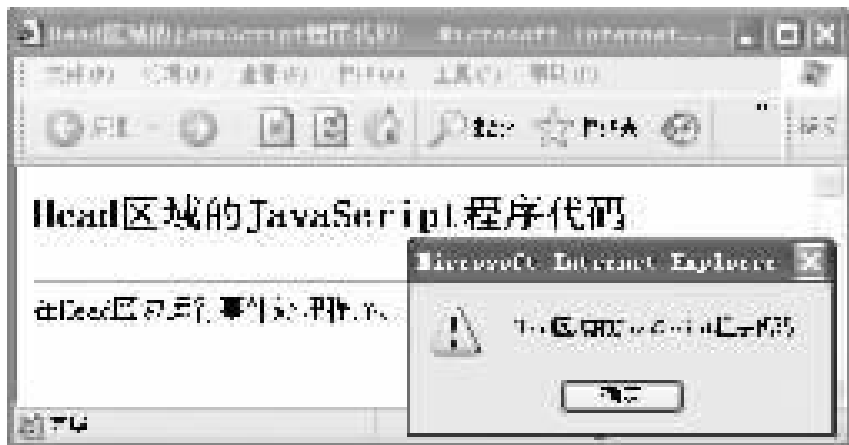


图 1-10 运行<head>标签中的 JavaScript 程序

由图 1-10 可以看出,在完全加载网页后,单击页面上的“在 Head 区域运行事件处理程序。”即触发 OnClick 事件,从而运行 alert 函数,显示对话框。单击对话框上的“确定”按钮,结束 JavaScript 程序代码的运行。

本章小结

本章介绍了 JavaScript 的历史、版本和特点,JavaScript 的 3 种编辑环境及 JavaScript 在 HTML 中的应用。

JavaScript 是一种基于对象的语言,有着丰富的对象和属性,后面章节的学习重点会集中在 JavaScript 的核心对象上。

习 题 1

一、简答题

1. 简述 JavaScript 的历史、版本和标准。
2. 简单说明 JavaScript 的 3 种编辑环境。
3. 简单介绍 JavaScript 在客户端和服务端的应用。

二、操作题

试编写 JavaScript 程序,实现在页面加载时弹出显示“大家好!”文本的对话框。