



高等职业教育电子信息类贯通制教材(计算机技术专业)

JavaScript 动态网页编程

◎ 赵增敏 主编

本书配有电子教学参考资料包



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

高等职业教育电子信息类贯通制教材（计算机技术专业）

JavaScript 动态网页编程

赵增敏 主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

JavaScript 是用于 Web 开发的主要脚本语言, 可以应用于客户端编程、服务器端编程以及 Ajax 应用开发。本书详细地介绍了 JavaScript 动态网页编程技术及其在 Web 开发中的应用。本书共分 11 章, 主要内容包括: JavaScript 使用基础、HTML 语言、CSS 样式表、JavaScript 语言基础、流程控制语句、面向对象编程、浏览器对象模型、文档对象模型、事件处理、表单与表单域、XMLHttpRequest 对象。本书结构合理、论述准确、内容翔实, 通过大量的实例深入浅出、循序渐进地引导读者学习, 每章后面均配有习题和上机实验。

本书既可作为中等、高等职业院校计算机相关专业或 JavaScript 培训班的教材, 也可作为 JavaScript 爱好者和动态网站开发维护人员的参考书。

本书还配有电子教学参考资料包(包括教学指南、电子教案和习题答案), 详见前言。

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 动态网页编程 / 赵增敏主编. —北京: 电子工业出版社, 2010.9

高等职业教育电子信息类贯通制教材. 计算机技术专业

ISBN 978-7-121-09642-6

I. J… II. 赵… III. JAVA 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 178998 号

策划编辑: 施玉新

责任编辑: 施玉新 特约编辑: 张荣琴

印 刷: 北京市李史山胶印厂

装 订: 北京市李史山胶印厂

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

开 本: 787×1092 1/16 印张: 19.5 字数: 528 千字

印 次: 2010 年 9 月第 1 次印刷

定 价: 32.00 元

凡所购买的电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

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。

服务热线: (010) 88258888。

前言



随着计算机网络技术的快速发展, Web 应用开发正在受到人们越来越多的关注。早期的 HTML 文档只能为访问者提供静态内容, 而不能生成动态效果和实现交互。JavaScript 语言正是在这种背景下应运而生的。

JavaScript 是一种基于对象和事件驱动并具有安全性能的脚本语言, 它是通过嵌入或导入到标准的 HTML 文档中实现的。把 JavaScript 脚本语言与 HTML、CSS 结合起来, 可以在一个网页中链接多个对象, 实现与网络客户的交互作用。

本书共分 11 章, 详细地讲述了使用 JavaScript 动态网页编程技术的基本知识和技巧, 并介绍了 JavaScript 在 Ajax 开发中的应用。第 1 章介绍什么是 JavaScript 以及如何在网页中编写 JavaScript 代码; 第 2 章讲述 HTML 网页设计, 主要包括 HTML 语言概述、设置文本格式、使用图像、添加媒体、创建表格、使用框架、使用超链接以及创建表单等; 第 3 章讨论如何使用 CSS 样式表来控制页面元素, 首先介绍 CSS 样式表的基本概念, 然后详细讲述各种 CSS 属性的设置; 第 4 章介绍 JavaScript 语言的基础知识, 包括基本语法、数据类型、变量和运算符等; 第 5 章介绍 JavaScript 流程控制语句, 主要包括条件语句、循环语句以及自定义函数等; 第 6 章讨论如何使用 JavaScript 实现面向对象编程, 主要包括对象基本概念、内置对象、本地对象以及用户自定义对象等; 第 7 章讨论各种 BOM 对象的应用, 主要包括 window 对象、document 对象、location 对象、navigator 对象以及 screen 对象等; 第 8 章讨论如何通过 DOM 处理 HTML 文档, 主要包括 DOM 概述、访问结点、操作结点以及处理表格等; 第 9 章讨论如何在 JavaScript 中进行事件处理, 主要包括设置事件属性、Internet Explorer 和 W3C DOM 的事件模型以及常用事件处理等; 第 10 章讨论如何使用 JavaScript 处理表单和表单域, 主要包括访问表单和表单域、文本框与文本区域、单选按钮与复选框以及列表框等; 第 11 章介绍 JavaScript 在 Ajax 开发中的应用, 讨论如何通过 XMLHttpRequest 对象以异步方式向服务器发送请求并接收和处理服务器响应, 主要包括 HTTP 请求/响应基础、使用 XMLHttpRequest 对象、发送异步请求以及处理服务器响应等。

本书由赵增敏主编。参加本书编写、代码测试和文字录入的还有朱粹丹、赵朱曦、余霞、刘家贵、王静、赵玉霞等, 在此一并致谢。

由于作者水平所限, 书中疏漏和错误之处在所难免, 欢迎广大读者提出宝贵意见。

为了方便教师教学, 本书还配有教学指南、电子教案和习题答案(电子版)。请有此需要的教师登录华信教育网(www.hxedu.com.cn)免费注册后进行下载, 有问题时请在网站留言板留言或与电子工业出版社联系(E-mail: hxedu@phei.com.cn)。

编 者



目 录



第 1 章	JavaScript 入门	(1)
1.1	JavaScript 概述	(1)
1.1.1	JavaScript 是什么	(1)
1.1.2	JavaScript 的特点	(2)
1.2	编写 JavaScript 代码	(2)
1.2.1	设置脚本语言	(3)
1.2.2	绑定对象事件	(4)
1.2.3	调用外部脚本库	(5)
1.2.4	通过属性调用脚本	(6)
1.2.5	处理不支持脚本的情况	(6)
习题 1		(8)
上机实验 1	JavaScript 使用基础	(8)
第 2 章	HTML 语言	(10)
2.1	HTML 语言概述	(10)
2.1.1	HTML 工作原理	(10)
2.1.2	HTML 网页基本结构	(11)
2.1.3	添加注释内容	(13)
2.2	设置文本格式	(13)
2.2.1	分段与换行	(13)
2.2.2	div 与 span 标记	(14)
2.2.3	设置字体、字号和颜色	(15)
2.2.4	设置字符样式	(16)
2.2.5	插入特殊字符	(17)
2.2.6	设置标题格式	(18)
2.2.7	设置列表格式	(19)
2.3	使用图像	(20)
2.3.1	在网页中插入图像	(20)
2.3.2	播放视频	(21)
2.3.3	插入水平分隔线	(22)
2.4	添加媒体效果	(23)
2.4.1	插入滚动字幕	(23)
2.4.2	插入声音效果	(25)
2.4.3	插入视频片段	(26)
2.4.4	插入 Flash 动画	(27)

2.4.5	插入 Applet	(27)
2.5	创建表格	(29)
2.5.1	创建基本表格	(29)
2.5.2	设置表格的属性	(30)
2.5.3	设置行的属性	(31)
2.5.4	设置单元格的属性	(32)
2.6	用框架分割窗口	(33)
2.6.1	框架网页的基本结构	(33)
2.6.2	设置框架集和框架的属性	(35)
2.7	用超链接实现导航	(36)
2.7.1	超链接概述	(37)
2.7.2	创建超链接	(37)
2.8	用表单实现交互	(39)
2.8.1	创建表单	(39)
2.8.2	创建输入型表单域	(40)
2.8.3	创建其他表单域	(42)
习题 2		(44)
上机实验 2	HTML 网页制作	(46)
第 3 章	CSS 样式表	(47)
3.1	CSS 样式表概述	(47)
3.1.1	什么是 CSS	(47)
3.1.2	CSS 选择符	(49)
3.1.3	CSS 单位	(51)
3.1.4	CSS 的应用	(52)
3.2	设置 CSS 属性	(53)
3.2.1	设置字体属性	(53)
3.2.2	设置背景属性	(57)
3.2.3	设置区块属性	(60)
3.2.4	设置方框属性	(62)
3.2.5	设置边框属性	(64)
3.2.6	设置列表属性	(68)
3.2.7	设置定位属性	(69)
3.2.8	设置扩展属性	(72)
习题 3		(75)
上机实验 3	使用 CSS 样式表设置网页外观	(76)
第 4 章	JavaScript 语言基础	(77)
4.1	基本语法	(77)
4.1.1	词法规则	(77)
4.1.2	关键字与保留字	(78)
4.2	数据类型	(78)
4.2.1	Number 类型	(79)
4.2.2	String 类型	(80)

4.2.3	Boolean 类型	(81)
4.2.4	特殊数据类型	(81)
4.2.5	数据类型转换	(82)
4.3	变量	(85)
4.3.1	变量的声明	(85)
4.3.2	变量的作用域	(86)
4.4	运算符	(87)
4.4.1	算术运算符	(88)
4.4.2	比较运算符	(89)
4.4.3	逻辑运算符	(90)
4.4.4	按位运算符	(91)
4.4.5	赋值运算符	(92)
4.4.6	其他运算符	(94)
4.4.7	运算符优先级	(95)
习题 4		(96)
上机实验 4 JavaScript 语言基础		(97)
第 5 章	流程控制语句	(98)
5.1	条件语句	(98)
5.1.1	if...else 语句	(98)
5.1.2	switch 语句	(99)
5.2	循环语句	(102)
5.2.1	do...while 语句	(102)
5.2.2	while 语句	(103)
5.2.3	for 语句	(104)
5.2.4	for...in 语句	(106)
5.2.5	跳转语句	(107)
5.3	异常捕获语句	(108)
5.3.1	try...catch 语句	(108)
5.3.2	throw 语句	(110)
5.4	函数	(111)
5.4.1	函数的定义和调用	(111)
5.4.2	参数数目可变的函数	(113)
5.4.3	匿名函数	(114)
5.4.4	嵌套函数	(115)
5.4.5	函数作为参数	(117)
习题 5		(118)
上机实验 5 流程控制语句应用		(118)
第 6 章	面向对象编程	(119)
6.1	对象基本概念	(119)
6.1.1	创建对象	(119)
6.1.2	对象类型	(120)
6.1.3	with 语句	(120)

6.2	内置对象	(121)
6.2.1	Global 对象	(121)
6.2.2	Math 对象	(124)
6.3	本地对象	(126)
6.3.1	Number 对象	(126)
6.3.2	String 对象	(128)
6.3.3	Boolean 对象	(134)
6.3.4	Array 对象	(135)
6.3.5	Date 对象	(140)
6.3.6	Object 对象	(144)
6.3.7	Function 对象	(146)
6.3.8	RegExp 对象	(148)
6.4	用户自定义对象	(150)
6.4.1	定义构造函数	(150)
6.4.2	对象直接量	(152)
6.4.3	实现继承机制	(153)
	习题 6	(155)
	上机实验 6 面向对象编程	(155)
第 7 章	浏览器对象模型	(157)
7.1	window 对象	(157)
7.1.1	window 对象概述	(157)
7.1.2	打开和关闭窗口	(159)
7.1.3	操作浏览器窗口	(161)
7.1.4	创建系统对话框	(162)
7.1.5	定时执行代码	(164)
7.1.6	导航浏览器历史	(166)
7.2	document 对象	(168)
7.2.1	document 对象概述	(168)
7.2.2	访问页面元素	(170)
7.2.3	向文档中写入内容	(171)
7.3	location 对象	(173)
7.3.1	location 对象概述	(174)
7.3.2	解析 URL	(175)
7.3.3	导航到其他网页	(176)
7.4	navigator 对象	(177)
7.4.1	navigator 对象概述	(178)
7.4.2	检测浏览器	(178)
7.5	screen 对象	(179)
7.5.1	screen 对象概述	(179)
7.5.2	检测客户端屏幕	(180)
	习题 7	(181)
	上机实验 7 浏览器对象模型应用	(181)

第 8 章 文档对象模型	(183)
8.1 DOM 概述	(183)
8.1.1 文档对象模型	(183)
8.1.2 文档节点树	(185)
8.1.3 节点的属性和方法	(186)
8.2 访问节点	(187)
8.2.1 访问根节点	(187)
8.2.2 访问相关节点	(188)
8.2.3 访问属性节点	(189)
8.2.4 访问指定节点	(192)
8.3 操作节点	(194)
8.3.1 创建节点	(194)
8.3.2 插入节点	(197)
8.3.3 复制节点	(198)
8.3.4 替换节点	(199)
8.3.5 移除节点	(201)
8.3.6 处理文本节点	(202)
8.4 处理表格	(204)
8.4.1 Table 对象	(204)
8.4.2 TableRow 对象	(208)
8.4.3 TableCell 对象	(209)
习题 8	(211)
上机实验 8 文档对象模型应用	(212)
第 9 章 事件处理	(213)
9.1 设置事件属性	(213)
9.1.1 在 JavaScript 中设置事件属性	(213)
9.1.2 在 HTML 中设置事件属性	(215)
9.1.3 使用代码触发事件	(217)
9.2 Internet Explorer 的事件模型	(219)
9.2.1 设置事件属性	(219)
9.2.2 事件对象	(221)
9.2.3 事件冒泡	(223)
9.2.4 事件转发	(224)
9.3 DOM 的事件模型	(225)
9.3.1 设置事件属性	(226)
9.3.2 事件对象	(228)
9.3.3 取消事件的默认行为	(230)
9.3.4 阻止事件的传播	(230)
9.4 常用事件	(232)
9.4.1 鼠标事件	(232)
9.4.2 键盘事件	(234)
9.4.3 HTML 事件	(235)

习题 9	(237)
上机实验 9 事件处理	(238)
第 10 章 表单与表单域	(239)
10.1 访问表单和表单域	(239)
10.1.1 访问表单	(239)
10.1.2 访问表单域	(240)
10.1.3 提交表单	(242)
10.1.4 重置表单	(245)
10.2 文本框与文本区域	(247)
10.2.1 文本框	(247)
10.2.2 文本区域	(250)
10.3 单选按钮与复选框	(252)
10.3.1 单选按钮	(252)
10.3.2 复选框	(254)
10.4 列表框	(257)
10.4.1 列表框概述	(257)
10.4.2 访问列表项	(259)
10.4.3 添加和删除列表项	(259)
习题 10	(263)
上机实验 10 表单与表单域应用	(264)
第 11 章 XMLHttpRequest 对象	(265)
11.1 HTTP 请求/响应基础	(265)
11.1.1 HTTP 请求	(265)
11.1.2 HTTP 响应	(267)
11.1.3 配置服务器端环境	(268)
11.2 使用 XMLHttpRequest 对象	(271)
11.2.1 创建 XMLHttpRequest 对象	(271)
11.2.2 XMLHttpRequest 对象的方法	(274)
11.2.3 XMLHttpRequest 对象的属性	(276)
11.3 发送异步请求	(279)
11.3.1 发送简单请求	(279)
11.3.2 发送 GET 请求	(281)
11.3.3 发送 POST 请求	(285)
11.3.4 发送 XML 请求	(288)
11.4 处理服务器响应	(291)
11.4.1 处理文本响应	(291)
11.4.2 处理 XML 响应	(294)
习题 11	(297)
上机实验 11 XMLHttpRequest 对象应用	(298)

第1章 JavaScript 入门

JavaScript 是用于 Web 开发的主要脚本语言, JavaScript 的主要用途包括: 通过客户端编程使开发者与网页和 Web 浏览器进行交互, 通过 Ajax 编程以异步方式向服务器发送请求并接收和处理服务器响应, 通过服务器端 ASP 编程来创建具有数据库访问功能的动态网站。本章首先对 JavaScript 及其特点做简单介绍, 然后讨论如何在网页中编写 JavaScript 代码。

1.1 JavaScript 概述

JavaScript 语言是 NetScape 公司为其 Web 浏览器 Navigator 开发的脚本语言, 其前身称为 LiveScript。自从 Sun 公司推出 Java 语言之后, Netscape 公司引进了有关 Java 语言的概念, 将原有的 LiveScript 重新进行设计并重命名为 JavaScript。

1.1.1 JavaScript 是什么

JavaScript 是一种基于对象和事件驱动并具有安全性能的脚本语言, 它是通过嵌入或导入到标准的 HTML 文档中实现的。把 JavaScript 脚本语言与 HTML、CSS 结合起来, 可以在一个网页中链接多个对象, 实现与网络客户的交互作用。

从名称上看, JavaScript 似乎与 Java 语言有很密切的关系。的确, JavaScript 语言与 Java 语言存在着一定的联系, 但它们之间的联系并不是想象中的那样紧密。JavaScript 与 Java 的区别体现在以下两个方面。

(1) 它们分别是两个公司开发的不同产品。JavaScript 是 Netscape 公司的产品, 其设计的旨在扩展 Netscape Navigator 浏览器的功能, 是一种可以嵌入网页中的基于对象和事件驱动的解释性语言; 而 Java 则是 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。

(2) JavaScript 是一种基于对象的脚本语言, 它可以嵌入 HTML 文档中, 用来实现用户与浏览器的交互以及浏览器与服务器的异步通信。Java 则是一种真正的面向对象的程序设计语言, 即使是开发简单的程序, 也必须基于类来创建对象。

ECMA (欧洲计算机制造协会) 于 1999 年在 JavaScript 1.5 的基础上制定了 ECMAScript 程序语言规范 (ECMA-262 标准), 该规范被 ISO (国际标准化组织) 采纳并作为浏览器使用的脚本程序的统一标准。

当 NetScape 公司把 JavaScript 成功地应用于 Navigator 浏览器之后, Microsoft 公司推出



了应用于 Internet Explorer 浏览器的脚本语言并将其命名为 JScript。JScript 与 JavaScript 的大部分功能是相同的,不过也有个别内容是不一样的。

当 ECMAScript 标准发布之后,NetScape 公司和 Microsoft 公司都承诺,以后为其浏览器脚本语言开发新的功能时都将遵循该标准。目前 JavaScript 的最新版本为 2.0, JScript 的最新版本为 5.7,两者都遵循尚未最后定稿的 ECMAScript-262 标准版本 4。

1.1.2 JavaScript 的特点

JavaScript 主要用于在 HTML 文档中编写客户端脚本代码。当用户在客户端浏览器中打开 HTML 网页时,浏览器就会执行该网页内嵌或导入的 JavaScript 代码,用户通过交互式操作更改网页的内容和外观,实现了仅用 HTML 和 CSS 所不能实现的效果。

概括地讲,JavaScript 具有以下特点。

(1) 简单性。JavaScript 是一种脚本编写语言,它采用小程序段的方式实现程序设计。与其他脚本语言一样,JavaScript 也是一种解释性语言,它提供了一个简单、方便的开发过程。JavaScript 的基本结构形式与 C、C++、Visual Basic、Delphi 十分类似,但不像这些语言一样需要事先进行编译,而是在程序运行过程中一边解释一边执行。JavaScript 与 HTML、CSS 结合在一起,可以实现用户的交互式操作。JavaScript 脚本语句的解释执行由 Web 浏览器负责,不需要额外的开发环境。

(2) 动态性。JavaScript 是动态的,它可以直接对用户操作做出响应,不需要经过 Web 服务器程序处理。

(3) 跨平台性。JavaScript 语言依赖于 Web 浏览器本身,而与操作环境无关,只要在计算机上能运行支持 JavaScript 的浏览器,就可以正确地执行 JavaScript 程序。

(4) 基于对象。在 JavaScript 脚本代码中,可以调用其自身提供的对象或其他语言创建的对象,并采用面向对象的编程方法来设置对象的属性、调用对象的方法,以完成所需功能。

(5) 事件驱动。JavaScript 对用户操作的响应是采用以事件驱动的方式进行的。所谓事件,通常是指在网页中执行了某种操作所产生的动作。例如,单击鼠标、移动窗口、选择菜单等都可以视为事件。当事件发生后,可能会引起事件响应,响应的具体内容则需要通过编写事件处理程序来实现。

(6) 节省与服务器端的交互时间。随着 Internet 的迅速发展,有许多 Web 服务器提供的服务需要与访问者进行交互,以检验访问者的身份及其可使用的服务内容等,这项工作通常由在服务器端编写相应的接口程序与用户进行交互来完成。很显然,通过网络与用户的交互过程不仅增大了网络的通信量,也影响了服务器的服务性能。JavaScript 是一种基于客户端浏览器的语言,当用户在浏览器中填写表单时,验证的交互过程可以通过浏览器对嵌入或导入 HTML 文档的 JavaScript 代码进行解释执行来完成,即使是必须调用服务器端验证程序,也可以通过 Ajax 技术以异步请求的方式把用户输入并通过验证的信息提交给远程服务器进行处理,从而极大地减小了服务器的开销。

1.2 编写 JavaScript 代码

在 HTML 文档中,JavaScript 代码可以直接包含在<script>与</script>标记之间,也可以

作为某个 HTML 元素的事件处理程序属性值,还可以把 JavaScript 代码存储在一个单独的文件(文件扩展名为.js)中并在网页的适当位置引用该文件。

1.2.1 设置脚本语言

编写 HTML 网页时,可以使用 script 标记提供一个容器,用于包含使用任何可由浏览器解释的脚本语言编写的代码行,以创建一个脚本代码块,语法如下:

```
<script language="javascript" type="text/javascript">  
    一个或多个脚本语句  
</script>
```

其中 language 属性指定 script 标记内代码行所使用的脚本语言,常用的值有 javascript、jscript 和 vbscript。对于 javascript,还可以指定脚本语言的版本如 javascript1.1、javascript1.2、javascript1.3、javascript1.4、javascript1.5。language 属性不再是 HTML 4 规范的一部分,如果编写 JavaScript 代码时需要进行 W3C 验证,则在 HTML 4.01 或 XHTML 1.0 中不验证此属性。

type 指定脚本语句的内容类型,以通知浏览器使用哪个脚本引擎来解释脚本语句。type 属性值为不区分大小写的 MIME 类型,常用值有 text/javascript、text/jscript、text/vbscript。当定义编写脚本语句所使用的脚本语言时,type 属性最终将代替 language 属性。由于早期浏览器忽略 type 属性,为了向前和向后兼容,建议同时包含 language 属性和 type 属性。

使用 script 标记时应注意以下几点。

- (1) 如果脚本语句未包含在一个函数定义内部,则这些脚本语句将在加载页面时执行。
- (2) 在脚本块中定义的函数在加载页面时将被加载,但其执行被延迟,直到用户或系统行为(事件)显式地调用该函数。
- (3) 在一个文档中可以有多多个 script 标记,而且这些元素中的代码可以相互访问。
- (4) 根据需要,可以把 script 标记放在 HTML 文档中的任何位置,既可以放在 head 元素内部,也可以放在 body 元素内部。
- (5) 在同一个文档中可以包含用不同脚本语言编写的 script 标记。

【例 1.1】 本例说明如何在 HTML 网页中创建一个 JavaScript 脚本块,用于显示“Hello, World!”问候语,网页运行结果如图 1.1 所示。

【代码清单】

```
<!-- F:\js\chapter01\page1-01.html -->  
<html>  
<head>  
<title>在网页中编写 JavaScript 脚本</title>  
</head>  
<body>  
<script language="javascript" type="text/javascript">  
    document.write("Hello, World!");  
</script>
```

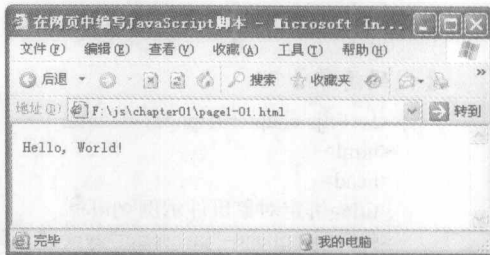


图 1.1 在网页中编写 JavaScript 脚本块



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

在浏览器中查看该网页的运行结果。

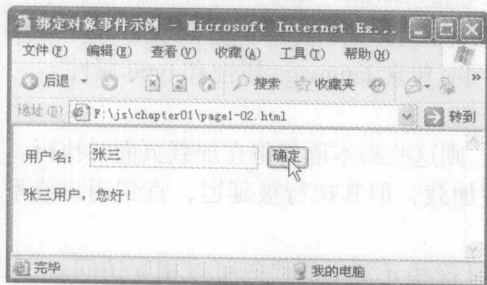
1.2.2 绑定对象事件

使用 Internet Explorer 浏览器的事件模型时, 允许通过 script 标记的 event、for 属性把某个对象与指定事件绑定起来, 语法如下:

```
<script language="javascript" type="text/javascript"
      for="elementID" event="eventName">
    事件处理程序代码
</script>
```

其中 for 属性的值为某个页面元素的 ID, event 属性的值则是该元素的事件名称, 该元素必须支持 event 属性指定的事件。

当加载页面时, 浏览器注册每个 script 标记并将其 event 属性指定的事件与 for 属性指定的页面元素绑定起来。当页面元素产生该事件时, 将执行 script 标记中的脚本语句, 不必对该对象编写事件处理程序, 或者把脚本语句包括在函数定义中。



注意:

只能在 Internet Explorer 浏览器中使用这种类型的脚本事件。Navigator 浏览器在加载页面时将试图执行脚本语句。

【例 1.2】 本例说明如何通过 for 属性和 event 属性把对象事件绑定到 script 标记。在文本框中输入用户名, 当单击“确定”按钮时将该用户显示问候信息, 如图 1.2 所示。

图 1.2 在网页中编写 JavaScript 脚本

【代码清单】

```
<!-- F:\js\chapter01\page1-02.html -->
<html>
<head>
<title>绑定对象事件示例</title>
<script language="javascript" type="text/javascript" for="btnOK" event="onclick">
    // 获取文本框的值
    var username = document.getElementById("txtUsername").value;
    // 获取段落的 ID
    var p1 = document.getElementById("p1");
    // 通过设置段落的 innerHTML 属性显示问候信息
    p1.innerHTML = username + "用户,您好!";
</script>
</head>
<body>
<p>用户名:
    <input type="text" name="txtUsername" id="txtUsername" />
    <input name="btnOK" type="button" id="btnOK" value="确定" />
```



```
</p>
<p id="p1">&nbsp;</p>
</body>
</html>
```

1.2.3 调用外部脚本库

通过把 script 标记的 src 属性设置为一个扩展名为.js 的外部脚本文件（称为脚本库）的路径，可以从外部文件中导入脚本语句。语法如下：

```
<script language="javascript" type="text/javascript" src="URL">
</script>
```

其中 src 属性指定外部 JavaScript 脚本文件的 URL。一旦加载外部语句，浏览器便对这些语句进行处理，就像把它们嵌入到 HTML 文档中一样。

在实际开发中，常常把一些通用功能编写为 JavaScript 函数，并且保存为扩展名为.js 的脚本库，然后在各个 HTML 文档顶部的 head 元素中导入该脚本库。当需要修改所有 HTML 文档中的函数定义时，对脚本库进行修改即可，十分便于实现网站的模块化设计。



注意：

在 JavaScript 脚本文件中，可以直接编写 JavaScript 语句，不需要把这些语句包含在 script 标记中。此外，当 script 标记包含 src 属性时，不要把其他脚本语句放在首尾标记之间。

【例 1.3】 本例说明如何通过 src 属性导入外部脚本文件。编写一个 JavaScript 脚本文件，然后在 HTML 网页中导入此脚本文件，网页运行结果如图 1.3 所示。

【代码清单】

```
// F:\js\chapter01\script1-03.js
function welcome(){           // 定义一个名为 welcome 的函数
    document.write("Now is " + new Date().toLocaleString() + "<br />");
    document.write("Welcome to JavaScript!");
}
<!-- F:\js\chapter01\page1-03.html -->
<html>
<head><title>导入外部脚本文件示例</title>
<script src="script1-01.js" language="javascript" type="text/javascript">
</script>
</head>

<body>
<p align="center">
<script language="javascript" type="text/javascript">
    welcome();                // 调用外部脚本文件中的函数
```

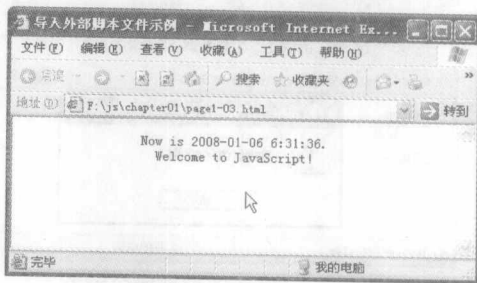


图 1.3 在网页中编写 JavaScript 脚本的运行结果

```

</script>
</p>
</body>
</html>

```

1.2.4 通过属性调用脚本

JavaScript 对标准 HTML 进行了扩展, 为 HTML 标记增加了各种事件属性, 通过设置这些事件属性可以调用 JavaScript 脚本。例如, 对于

若要通过 a 标记的 href 属性调用 JavaScript 代码, 则必须在用做 URL 的 JavaScript 代码之前添加 “javascript:”。也可以将 href 属性设置为 “#”, 而在事件属性 onclick 中设置要执行的 JavaScript 代码, 此时不需要在代码之前添加 “javascript:”, 通过在 JavaScript 代码中添加一个 “return false;” 语句, 则可以取消这个超链接的跳转作用。

【例 1.4】 本例说明如何通过 HTML 属性设置要执行的 JavaScript 代码, 网页运行结果如图 1.4 和图 1.5 所示。

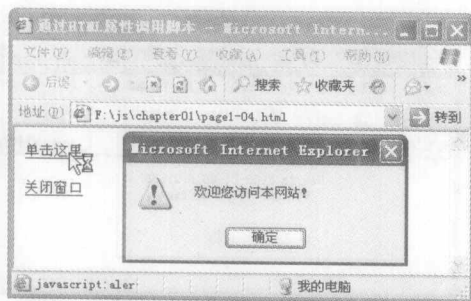


图 1.4 通过 href 属性调用脚本



图 1.5 通过 onclick 属性调用脚本

【代码清单】

```

<!-- F:\js\chapter01\page1-04.html -->
<html>
<head>
<title>通过 HTML 属性调用脚本</title>
</head>

<body>
<!-- 单击 “单击这里” 链接时将弹出一个对话框, 用于显示欢迎信息 -->
<p><a href="javascript:alert('欢迎您访问本网站! ? ');">单击这里</a></p>
<!-- 单击 “关闭窗口” 链接时将弹出一个对话框, 提示关闭当前浏览器窗口 -->
<p><a href="#" onclick="window.close(); return false;">关闭窗口</a></p>
</body>
</html>

```

1.2.5 处理不支持脚本的情况

某些浏览器不能很好地支持 JavaScript, 会把 JavaScript 代码与网页中的其他 HTML 内

容一起显示在浏览器窗口中。在这种场合,可以使用 HTML 注释标记 (`<!--`和`-->`) 把脚本块中的 JavaScript 语句括起来,以避免显示这些脚本代码,语法如下:

```
<script language="javascript" type="text/javascript">
<!--
    被屏蔽的 JavaScript 语句
-->
</script>
```

支持 script 标记的浏览器将忽略包含 `<script>` 与 `</script>` 之间的 HTML 注释标记,而不会影响脚本语句的解释执行。

对于不支持或禁止运行 JavaScript 脚本的浏览器,可以使用 `noscript` 标记指定要显示的 HTML 内容,语法如下:

```
<noscript>
    要显示的 HTML 内容
</noscript>
```

如果一个网页中同时包含 script 标记和 noscript 标记,则在支持 JavaScript 的浏览器中加载该网页时,将执行 script 标记中的脚本代码并显示其运行结果;在不支持或阻止脚本的浏览器中加载该网页时,会把 noscript 标记指定的内容显示出来,但 script 标记和 noscript 标记指定的内容不会同时出现。

【例 1.5】 本例说明浏览器如何处理不支持脚本的情况。当在浏览器中加载网页时,在地址栏下方出现了一个提示条,并在浏览器窗口中显示由 noscript 标记指定的内容,如图 1.6 所示;如果单击该提示条并从弹出菜单选择“允许阻止的内容”命令,然后在“安全警告”对话框中单击“是”按钮,如图 1.7 所示,则由 noscript 标记指定的内容被隐藏起来,取而代之的是执行 JavaScript 脚本所生成的内容,如图 1.8 所示。

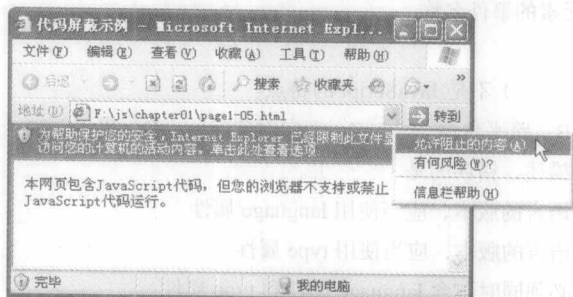


图 1.6 JavaScript 脚本被阻止的情形

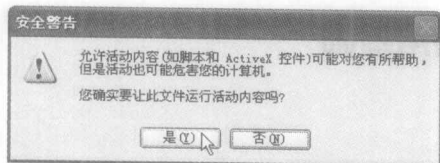


图 1.7 允许浏览器运行脚本

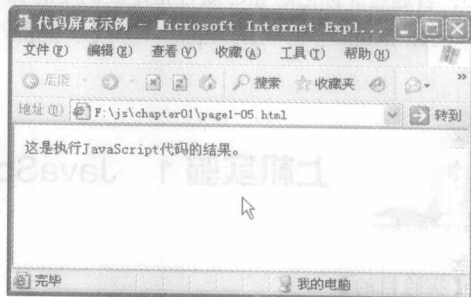


图 1.8 执行脚本的结果