



普通高等教育“十一五”规划教材

高职高专计算机软件技术专业系列教材

```
<CENTER><B><FONT SIZE 12>My First Web Page </FONT></B></CENTER>
<P>I am Learning HTML 1. I <B>I LOVE</I></B>HTML!
<P><CENTER><IMG SRC 1advjscript.jpg></CENTER>
<P>
<P><CENTER>You can email me at</CENTER>
<P>
<CENTER><A HREF 1mailto:myemail@someemail.com>Email ME</A>
</CENTER>
<P><CENTER>Or go to this publisher's Web Site </CENTER>
<P><CENTER><A HREF 1http://www.aardware.com>Aardware Publishing</A></CENTER>
</BODY>
</HTML>
```

HIGHER TECHNICAL
AND
VOCATIONAL
EDUCATION

JavaScript

程序设计

刘怀亮 主编

黄常青 胡品辉 编著

冶金工业出版社

普通高等教育“十一五”规划教材
高职高专计算机软件技术专业系列教材

JavaScript 程序设计

刘怀亮 主编

黄常青 胡品辉 编著

北 京

冶金工业出版社

内 容 简 介

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

本书全面系统地介绍了用 JavaScript 语言制作网页的编程技术方法。主要内容包括：JavaScript 编程基础，变量、数据类型和表达式，流程控制，函数，对象编程，文档对象模型与事件驱动，窗口、文档和表单对象，级联样式表（CSS）技术，动态对象和 Cookies 及安全等。附录 A 中针对书中的主要内容分别提供了 10 个典型的操作实训。

本书由浅入深地介绍了 JavaScript 的语法，并且对每个语法都举出了典型的例子，使读者能够迅速地理解语法的内容，并能融入到实际的操作中。本书既强调理论性又注重实践，并以各类实例引导读者全面掌握 JavaScript 编程的方法和技巧。

本书主要满足高职高专院校计算机及相关专业的教学需求，也可作为高职高专院校非计算机专业高年级的选修教材。对网站设计、网页制作感兴趣并且有一定理论基础的人员也可通过本书进行深入学习，同时也可供从事与计算机科学与技术有关的科技工作者参考。

图书在版编目（CIP）数据

JavaScript 程序设计 / 刘怀亮主编；黄常青，胡品辉
编著. —北京：冶金工业出版社，2006.11
ISBN 7-5024-4136-0

I. J... II. ①刘...②黄...③胡... III. JAVA 语言—
程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2006）第 128675 号

出版人 曹胜利（北京沙滩嵩祝院北巷 39 号，邮编 100009）

责任编辑 戈兰

广州锦昌印务有限公司印刷；冶金工业出版社发行；各地新华书店经销

2006 年 11 月第 1 版第 1 次印刷

787mm×1092mm 1/16； 13 印张； 296 千字； 200 页

20.00 元

冶金工业出版社发行部 电话：（010）64044283 传真：（010）64027893

冶金书店 地址：北京东四西大街 46 号（100711） 电话：（010）65289081

（本社图书如有印装质量问题，本社发行部负责退换）

前 言

一、关于本书

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

JavaScript 是一种语法简单但功能较强大的脚本编程语言。作为一种脚本语言，JavaScript 已经被广泛地运用到网页制作和网站设计中，并且得到各种主流 Web 浏览器的支持。

本书作者依据计算机专业的相关教学大纲，通过大量工程实践，在教学实践中编写了这本书，现将它奉献给广大读者。

二、本书结构

本书全面系统地介绍了用 JavaScript 语言制作网页的编程技术方法。全书共 12 章，具体结构安排如下：

第 1 章：JavaScript 语言概论。介绍了 JavaScript 语言的特点、作用，熟悉编辑工具，并编写第一个 JavaScript 程序。

第 2 章：JavaScript 代码编写与调试。介绍了 JavaScript 代码的编写与调试方法、JavaScript 的语法规则、代码的调试方法与技巧。

第 3 章：与用户交互方法。介绍了比较常用的人机交互方法——消息对话框。具体介绍了消息对话框的三种特殊的形式——警告对话框、确认对话框和提示对话框的生成和使用方法。

第 4 章：变量、数据类型和表达式。介绍了变量、数据类型、表达式的概念和使用方法，变量的声明、命名、初始化和赋值，各种数据类型的概念和不同数据类型间的转换方法，表达式的计算方法。

第 5 章：JavaScript 控制语句。介绍了 JavaScript 的流程控制语句，主要包括条件控制语句和循环控制语句的使用。

第 6 章：JavaScript 函数。介绍了函数的定义、声明和各种使用方式，如何利用多种方式创建和使用函数，理解函数参数的传递、函数返回值和函数的作用域。

第 7 章：JavaScript 对象编程。详细地介绍了 JavaScript 对象的概念、对象的功能和对象间的关系。

第 8 章：文档对象模型和事件驱动。介绍了 JavaScript 的动态网页编程技术。

第 9 章：窗口、文档和表单对象。介绍了窗口对象、文档对象和表单对象。

第 10 章：级联样式表 (CSS)。介绍了在 HTML 页面中定义和使用样式的方法，CSS 的常用属性以及 JavaScript 对这些属性的处理方法。

第 11 章：动态对象。介绍了动态 HTML、文档对象模型、Image 对象、使用 Image 对象的动画、图像缓冲以及 CSS 在 Internet Explorer 和 Navigator 中的使用。

第 12 章：Cookies 及安全。介绍了 Cookies 的概念和基本的使用方法，以及在网络安

全方面与 JavaScript 有关的内容、同源策略、签署脚本和数字证书。

附录 A 中针对书中的主要内容分别提供了 10 个典型的操作实训，给出了上机实训和操作过程，使读者在理解基本内容的同时进一步的加强实践操作。

附录 B 列出了几个关于 JavaScript 语言的学习网站，以供读者参考。

三、本书特点

本书内容覆盖了 JavaScript 的各知识要点，并对重点进行了详细的阐述。做到了内容新颖、概念清晰、讲究实际、通俗易懂、循序渐进。

本书由浅入深地介绍了 JavaScript 的语法，并且对每个语法点都举出了典型的例子，使读者能够迅速地理解语法的内容，并能融入到实际的操作中。

本书既强调理论又注重实践，并以各类实例引导读者全面掌握 JavaScript 编程的方法和技巧，本书的例子旨在针对性地实现网页的某种特效，强调在制作网络项目中的实战效果。

为方便读者进一步的实践，附录 A 中针对书中的主要内容分别提供了 10 个典型的操作实训，给出了上机实训和操作过程，使读者在理解基本内容的同时进一步加强实践操作。

四、适用对象

本书主要满足高职高专院校计算机及相关专业的教学需求，也可作为高职高专院校非计算机专业高年级的选修教材，对网站设计、网页制作感兴趣并且有一定理论基础的人员也可通过本书进行深入学习，同时也可供从事计算机科学与技术相关工作的科技工作者参考学习。

由于编写时间仓促，水平有限，书中疏漏和不足之处在所难免，恳请广大专家和读者朋友批评指正。联系方法如下：

电子邮箱：great_liu@126.com

网址：www.cnbook.net

本书电子教案、代码以及习题参考答案可从该网站下载，此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编者
2006 年 8 月

目 录

第 1 章 JavaScript 语言概论 1	
1.1 JavaScript 语言概况 1	
1.1.1 JavaScript 语言的特点 1	
1.1.2 JavaScript 可以做什么 2	
1.1.3 JavaScript 语言和其他语言的 比较 2	
1.1.4 JavaScript 的执行平台 4	
1.2 在 Web 页面中使用 JavaScript 4	
1.2.1 HTML 的基本结构 5	
1.2.2 在 HTML 中嵌入 JavaScript 5	
1.2.3 链接 JavaScript 文件 6	
1.3 编写 JavaScript 的工具 6	
1.3.1 使用纯文本编辑器 6	
1.3.2 使用专业化脚本编辑工具 7	
1.3.3 使用 Microsoft 脚本编辑器 7	
1.4 第一个 JavaScript 程序 9	
1.4.1 关于 < SCRIPT > 标签 9	
1.4.2 为 JavaScript 程序添加注释 10	
1.4.3 在 < HEAD > 或 < BODY > 段 放置 JavaScript 10	
小结 11	
习题一 11	
一、选择题 11	
二、填空题 11	
三、思考题 11	
四、上机操作题 11	
第 2 章 JavaScript 代码编写与调试 12	
2.1 JavaScript 代码的编写和执行 12	
2.2 语法规则 12	
2.2.1 大小写 12	
2.2.2 代码书写格式 13	
2.2.3 保留字 14	
2.2.4 分号和语句结束符 14	
2.3 动态生成页面内容 14	
2.3.1 字符串及其拼接方法 14	
2.3.2 document 对象 write()和 writeln()方法 15	
2.4 代码的调试方法与技巧 16	
2.4.1 错误类型 16	
2.4.2 调试工具 17	
2.4.3 调试技巧 18	
小结 19	
习题二 19	
一、选择题 19	
二、填空题 19	
三、思考题 19	
四、上机操作题 19	
第 3 章 与用户交互方法 20	
3.1 消息对话框 20	
3.2 显示警告对话框的 alert()方法 20	
3.2.1 警告对话框 20	
3.2.2 alert()方法及使用 21	
3.3 显示确认消息框的 confirm()方法 21	
3.3.1 确认信息框 21	
3.3.2 confirm()方法及使用 22	
3.4 显示提示对话框的 prompt()方法 23	
3.4.1 提示对话框 23	
3.4.2 prompt()方法 23	
小结 24	
习题三 24	
一、选择题 24	
二、填空题 24	
三、思考题 25	
四、上机操作题 25	
第 4 章 变量、数据类型和表达式 26	
4.1 变量 26	
4.1.1 变量声明和变量初始化 26	

4.1.2 变量的命名	26	小结	48
4.1.3 变量的作用域	26	习题五	48
4.2 数据类型和数组	27	一、选择题	48
4.2.1 数据类型	27	二、填空题	49
4.2.2 字符串型	27	三、思考题	49
4.2.3 数值型	28	四、上机操作题	49
4.2.4 布尔型	29	第 6 章 JavaScript 函数	50
4.2.5 数组	29	6.1 什么是函数	50
4.3 数据类型转换	30	6.2 定义函数和使用函数	50
4.4 表达式和运算符	30	6.3 函数参数的传递	54
4.4.1 表达式	30	6.4 在多个位置调用函数	54
4.4.2 算术运算符和赋值运算符	31	6.4.1 在事件中调用函数	54
4.4.3 比较运算符和逻辑运算符	32	6.4.2 通过链接调用函数	56
4.4.4 位运算符和条件运算符	33	6.5 函数中变量的作用域与函数的 返回值	57
4.4.5 其他运算符	34	6.5.1 函数中变量的作用域	57
4.4.6 运算符优先级	35	6.5.2 函数的返回值	59
小结	36	6.6 函数的高级用法——递归与嵌套	60
习题四	36	6.6.1 递归函数	60
一、选择题	36	6.6.2 函数的嵌套	61
二、填空题	36	小结	62
三、思考题	37	习题六	62
四、上机操作题	37	一、选择题	62
第 5 章 JavaScript 控制语句	38	二、填空题	63
5.1 条件控制语句	38	三、思考题	63
5.1.1 if 语句	38	四、上机操作题	63
5.1.2 if...else 语句	39	第 7 章 JavaScript 对象编程	64
5.1.3 switch 语句	41	7.1 初探对象编程	64
5.1.4 条件运算符	42	7.2 对象的基本概念	65
5.2 循环语句	43	7.2.1 什么是对象	65
5.2.1 while 语句	43	7.2.2 在 JavaScript 中使用对象	65
5.2.2 do...while 语句	44	7.2.3 创建和删除对象	66
5.2.3 for 语句	44	7.2.4 对象的属性和方法	66
5.3 与对象有关的语句	45	7.3 Math 对象	67
5.3.1 for...in 语句	45	7.4 Date 对象	69
5.3.2 with 语句	46	7.4.1 Date 对象的主要方法	69
5.4 循环控制	46	7.4.2 创建 Date 对象	72
5.4.1 continue 语句	46		
5.4.2 标签	47		

7.5 Number 对象.....	72	9.2.1 document 对象的属性和方法	100
7.6 String 对象.....	73	9.2.2 使用 all 属性访问 HTML 元素 ...	102
7.7 Array 对象	75	9.3 表单 (form) 对象.....	104
7.7.1 创建和访问数组.....	75	9.3.1 form 对象的属性、方法和	
7.7.2 Array 对象的常用属性和方法	76	事件	104
7.7.3 二维数组	78	9.3.2 表单处理	106
小结	79	9.3.3 表单元素对象的属性、方法和	
习题七	79	事件	109
一、选择题	79	9.3.4 处理表单元素示例	111
二、填空题	80	小结	114
三、思考题	80	习题九.....	114
四、上机操作题	80	一、选择题	114
第 8 章 文档对象模型和事件驱动	81	二、填空题	115
8.1 文档对象模型	81	三、思考题	115
8.1.1 认识文档对象模型	81	四、上机操作题	115
8.1.2 引用文档对象模型中的对象	81	第 10 章 级联样式表 (CSS)	116
8.2 事件驱动	82	10.1 CSS 简介	116
8.2.1 事件驱动的基本概念	82	10.2 样式定义	116
8.2.2 JavaScript 的常用事件	82	10.2.1 样式定义的格式	116
8.3 处理事件	84	10.2.2 选择器的分类	117
8.3.1 使用事件的方法	84	10.3 使用样式	118
8.3.2 使用制作工具自动生成		10.3.1 嵌入样式表	118
事件处理函数的模板	87	10.3.2 链接外部样式表	118
8.4 使用事件 (Event) 对象.....	88	10.3.3 内嵌样式	120
小结	90	10.3.4 CSS 样式的优先级	120
习题八	90	10.4 Style 对象.....	121
一、选择题	90	10.5 CSS 属性	122
二、填空题	90	10.5.1 字体属性	122
三、思考题	90	10.5.2 文本属性	123
四、上机操作题	90	10.5.3 颜色和背景属性	124
第 9 章 窗口、文档和表单对象	91	10.5.4 容器属性	127
9.1 窗口 (window) 对象	91	10.5.5 列表属性	131
9.1.1 window 对象的属性和方法	91	10.5.6 鼠标属性	134
9.1.2 打开和关闭窗口	92	10.5.7 定位和显示	135
9.1.3 使用定时器	94	10.5.8 CSS 滤镜	138
9.1.4 页面跳转	97	小结	143
9.2 文档 (document) 对象	100	习题十.....	143
		一、选择题	143

二、填空题	143	12.2.1 同源策略	161
三、思考题	144	12.2.2 签署脚本和数字证书	161
四、上机操作题	144	12.3 JavaScript 安全	163
第 11 章 动态对象	145	12.3.1 JavaScript 安全所关心的 内容	163
11.1 动态对象模型	145	12.3.2 编写安全的 JavaScript 程序	163
11.1.1 动态 HTML	145	小结	164
11.1.2 文档对象模型	145	习题十二	164
11.1.3 Image 对象	147	一、选择题	164
11.1.4 使用 Image 对象的动画	149	二、填空题	165
11.1.5 图像缓冲	151	三、思考题	165
11.2 动画和层叠式表单	153	四、上机操作题	165
11.2.1 层叠式表单	153	附录 A 上机实训	166
11.2.2 在 JavaScript 中使用 CSS 样式	154	实训 1 JavaScript 基本编程工具熟悉、 代码调试	166
11.2.3 CSS 定位	155	实训 2 JavaScript 常用内置对象	167
11.2.4 在 Internet Explorer 中定位	156	实例 3 JavaScript 计算器	170
11.2.5 在 Navigator 中定位	156	实训 4 JavaScript 控制语句	173
小结	156	实训 5 JavaScript 函数	175
习题十一	157	实训 6 JavaScript 对象	181
一、选择题	157	实训 7 JavaScript 文档对象模型和 事件驱动	188
二、填空题	157	实训 8 JavaScript 窗口、文档和 表单检验	189
三、思考题	157	实训 9 JavaScript 实时日历	193
四、上机操作题	157	实训 10 JavaScript 制作解释型菜单	197
第 12 章 Cookies 及安全	158	附录 B 学习资源网站	199
12.1 Cookie	158	参考文献	200
12.1.1 使用 Cookie 保存状态信息	158		
12.1.2 创建和设置 Cookie	159		
12.2 网络安全策略	161		

第 1 章 JavaScript 语言概论

本章教学目标：

JavaScript 是 Web 页面中的一种脚本编程语言，可用于 Web 系统的客户端和服务端编程，通过本章的学习，读者可以了解 JavaScript 语言的特点、作用，熟悉编辑工具，并能编写简单的 JavaScript 程序。

1.1 JavaScript 语言概况

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。使用它的目的是与 HTML 超文本标记语言、Java 脚本语言 (Java 小程序) 一起实现在一个 Web 页面中链接多个对象，与 Web 客户交互作用，从而可以开发客户端的应用程序等。它是通过嵌入或调入在标准的 HTML 语言中实现的。

1.1.1 JavaScript 语言的特点

JavaScript 的出现弥补了 HTML 语言的缺陷，是 Java 与 HTML 折衷的选择，它具有以下几个基本特点：

1. 一种脚本编程语言

首先要解释一下什么是脚本语言，脚本语言是一种简单的程序，由一些 ASCII 字符构成，它直接利用记事本等文本编辑软件就可以开发完成，并且事先无需编译，只要利用适当的解释器就可以转译并执行。

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 同样是一种解释性语言，它提供了一个容易的开发过程。其基本结构形式与 C、C++、VB、Delphi 十分类似。但它不像这些语言需先进行编译然后才能执行，而是在程序运行过程中被逐行地解释执行。它与 HTML 标识结合在一起，从而方便了用户的使用操作。

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创建的对象。因此，许多功能可以来自于脚本环境中对象方法调用。

3. 简单性

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；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。

4. 安全性

JavaScript 是一种安全的语言，它不允许访问本地的硬盘，且不能将数据存入到服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互，从而使对数据的操作安全化。

5. 动态性

JavaScript 是动态的，它可以直接对用户或客户的输入做出响应，无需经过 Web 服务

程序。它对用户的响应，是采用事件驱动的方式进行的。所谓事件驱动，是指在页面中执行了某种操作后产生相应的动作，比如按下鼠标、移动窗口、选择菜单等都可以视为事件，而当事件发生后，可能会引起相应的事件响应。

6. 跨平台性

JavaScript 与操作环境无关，只依赖于浏览器本身，只要计算机支持实现了 JavaScript 的浏览器，它就可被正确执行。从而实现了“编写一次，走遍天下”的梦想。

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（如 form 的输入），而不用任何的网路来回传资料，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。

1.1.2 JavaScript 可以做什么

JavaScript 虽然是一种简单的语言，但它的功能却很强大，具体如下：

1. 制作网页特效

这几乎是所有 JavaScript 的初学者都想学习 JavaScript 的第一个动机，例如：光标动画、信息提示、动态广告面板、检测鼠标行为等。

2. 提升使用性能

越是复杂的代码，越要耗费系统资源来执行它，因为大部分的 JavaScript 程序代码都在客户端执行，操作时完全不用服务器操心，这样网页服务器就可以将资源用在提供客户端更多更好的服务上。现今，越来越多的网站包含表单的结构，例如：申请会员要填写入会的基本表单，JavaScript 的任务就是在 CGI 程序将客户端所填写的数据送到服务器之前，先作必要的资料有效性测试，例如：该输入数字的地方是否有数字等等，这样的验证无疑提升了性能。

3. 窗口动态操作

利用 JavaScript，可以很自由地设计网页窗口的大小、窗口的打开与关闭等，甚至可以在不同窗口文件中互相传递参数。

1.1.3 JavaScript 语言和其他语言的比较

1. JavaScript 和 Java 的区别

Java 是由 Sun 公司开发的一种与平台无关的面向对象程序设计语言，特别适合于 Internet 应用程序开发，它与 JavaScript 基于对象的结构相反。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 可以用来设计独立的应用程序，也可以用来创建一种称为 Applets 的小应用程序。经过编译后，Applets 成为一种与平台无关的字节代码，这种 Applets 可以运行在任何平台上，只要该平台具备能够操作 Applets 的 Java 虚拟机即可。这样，开发人员可以摆脱为某个特定的平台编译代码的束缚。下面对这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 是一种脚本语言，它可以用来制作与网络无关的，与用户交互作用的复杂软件。它是一种基于对象和事件驱动的编程语言。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，即所有变量在编译之前必须作声明。如：

```
Integer a;  
String b;  
a=33;  
b=44;
```

其中前者定义 a 为整型变量，后者定义 b 为字符型变量。

JavaScript 中变量声明，采用弱类型。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=33;  
b="44";
```

前者说明 a 为数值型变量，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，必须通过像在 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 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，两种编程语言的标识不同，JavaScript 使用<Script>和</Script>来标识，而 Java 使用<applet>和</applet>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，如不经编译则就无法实现对象引用的检查。

2. JavaScript 和 VBScript 的区别

VBScript 是 Microsoft 公司开发的 Visual Basic 程序设计语言家族的新成员。它是 Visual Basic 轻量级的子集。与 JavaScript 相同，它被嵌入在 HTML 主页中，当浏览器检索到它时对其进行解释。使用 VBScript，可以验证表单数据的合法性，动态产生用户主页，甚至无需进行服务器方编程，就可以直接在 Web 页面内编写小游戏程序。VBScript 被设计用来与 ActiveX 控件和嵌入在 HTML 主页中的其他对象协同工作。另外，与 JavaScript 不同，VBScript 还可以作为普通的脚本语言在其他的应用程序中使用。

3. JavaScript 和 Livewire 的区别

Livewire 与 Livewire Pro 是 Netscape 发行的新产品的集合。Livewire 不仅仅是一种脚本语言，尽管它是同 Java 兼容的脚本语言，但这只是它功能的一小部分。Livewire 包括一个所见即所得的编辑浏览器，一个图形 Web 节点管理器。在 Pro 的版本中，它还支持 Informix、Oracle、Sybase 和 Microsoft 公司的 SQL 数据库。Livewire 只能运行在 Netscape 服务器上。

目前，JavaScript 并不支持数据库的直接存取操作。但在不久的将来，应该可以使用 JavaScript 来创建并操作大型数据库。

4. JavaScript 和 Perl 的区别

Perl 是一种解释型语言，它主要用于读取、解释和实现打印正文文件。常常被用于处理 E-mail 的请求与生成。Perl 的语言形式与使用类似于 C 语言，它也可以将一个很大的文本文件读入到一个单个的字符串中。可以使用 Perl 来读取格式数据，并将信息包装为可读的格式返回到用户。Perl 也可以返回一个动态创建的 HTML 主页。除此之外，由于 Perl 是运行在服务器而不是客户机上，它还可以对服务器的驱动器进行读、写操作。

目前，JavaScript 尚不能进行正文文件的操作，它也不能对输入的信息流进行分析。

1.1.4 JavaScript 的执行平台

JavaScript 既然是 Netscape 公司发展出来的脚本语言，当然可以在 Netscape 公司自家的浏览器软件 Navigator 中执行，因此从早期的版本(Navigator 2.0)到推出不久的(Navigator 6.0)都全力支持 JavaScript。

微软从 IE 3.0 以后的版本开始陆续地支持不同版本的 JavaScript，有了微软作为坚强后盾，JavaScript 马上走红，迅速席卷网页脚本语言的市场。

时至今日，几乎所有的浏览器都已经支持 JavaScript。因为不支持的几乎都已经被 Internet 市场所淘汰。

1.2 在 Web 页面中使用 JavaScript

经过前面的讲述，想必大家也已经知道，JavaScript 是内嵌在 HTML 中被执行的，在讲解如何在网页中使用 JavaScript 进行程序设计之前，先向大家介绍一些背景知识，让大家有一个大概的了解。

HTML 的全写是：Hypertext Marked Language，即超文本标记语言，是一种用来制作超文本文档的简单标记语言。用 HTML 编写的超文本文档称为 HTML 文档，它能独立于各种操作系统平台（如 UNIX、Windows 等）。自 1990 年以来 HTML 就一直被用作 World Wide Web 的信息表示语言，用于描述 Homepage 的格式设计与与 WWW 上其他 Homepage 的连结信息。使用 HTML 语言描述的文件，需要通过 WWW 浏览器显示出效果。

所谓超文本，它可以加入图片、声音、动画、影视等内容，它可以从一个文件跳转到另一个文件，与世界各地主机的文件连接。如：

通过 HTML 可以表现出丰富多彩的设计风格。

图片调用：

文字格式：文字

通过 HTML 可以实现页面之间的跳转。

页面跳转: ``

通过 HTML 可以展现多媒体的效果

音频: `<EMBED SRC="音乐文件名" AUTOSTART=true>`

视频: `<EMBED SRC="视频文件名" AUTOSTART=true>`

1.2.1 HTML 的基本结构

超文本文档分文档头和文档体两部分, 在文档头里, 对这个文档进行了一些必要的定义, 文档体中才是要显示的各种文档信息。

先来看一个简单的例子, 下面这个例子给出了 HTML 的基本结构。

```
<HTML>
<HEAD>
<TITLE>HTML 的基本结构</TITLE>
</HEAD>
<BODY>
<P>一个 HTML 的基本结构的例子</P>
</BODY>
</HTML>
```

这个例子的结果如图 1-1 所示。



图 1-1 HTML 的基本结构

由这个例子可以看出, HTML 的基本结构有几个特征。

- (1) HTML 文件由一个 `<HTML>` 元素组成, 即文件以 `<HTML>` 开始, 以 `</HTML>` 结束。
- (2) `<HTML>` 由头元素 `<HEAD>` 和体元素 `<BODY>` 组成。
- (3) 头元素和体元素又由其他元素、文本及注释组成。

1.2.2 在 HTML 中嵌入 JavaScript

在 HTML 中通过标记 `<SCRIPT>……</SCRIPT>` 引入 JavaScript 代码, 当浏览器读取到 `<SCRIPT>` 标记时, 就解释执行其中的脚本。

下面通过一个例子, 编写第一个 JavaScript 程序。通过它可说明 JavaScript 的脚本是怎样被嵌入到 HTML 文档中的。源程序如下:

```
<html>
<head>
<Script Language = "JavaScript">
// JavaScript 代码放在这里.
alert("在 HTML 中嵌入 JavaScript!");
</Script>
</Head>
```

```
</html>
```

如同 HTML 标识语言一样，JavaScript 程序代码是一些可用字处理软件浏览的文本，它出现在描述页面的 HTML 相关区域。

1.2.3 链接 JavaScript 文件

在 Web 页中引入 JavaScript 程序的另一种形式是采用链接 JavaScript 文件的形式。如果脚本程序较长或者同一段脚本可以在若干个 Web 页中使用，则可以将脚本放在单独的一个 .js 文件里，然后链接到需要它的 HTML 文件，这相当于将其中的脚本填入链接处。

要引用外部脚本文件，使用<SCRIPT>标记的 SRC 属性来指定外部脚本文件的 URL。如果使用了<SCRIPT>标记的 SRC 属性，则 Web 浏览器只使用在外部文件中的脚本，并忽略位于该<SCRIPT>标记之间的任何脚本。

具体做法如下：

(1) 使用文件编辑器编写一个 JavaScript 脚本程序，其中不需要任何 HTML 标记（包括<SCRIPT>标记），而仅仅有 JavaScript 代码。

(2) 将该文件以扩展名 .JS 保存在与要链接的 HTML 文件相同的位置上。

(3) 在 HTML 文件中使用<SCRIPT>标记的 SRC 属性来链接该 .js 文件。

下面的例子说明了如何链接 JavaScript 代码。源程序如下：

```
<HTML><HEAD>
<TITLE>链接 JavaScript 代码</TITLE>
</HEAD>
<BODY>
<SCRIPT language="JavaScript" src="s01_03.js">
</SCRIPT>
</BODY></HTML>
```

其中，文件 s01_03.js 与文件 s01_03.htm 处于相同的文件夹中，并含有如下内容：

```
document.write("hello world^_^!");
```

结果如图 1-2 所示。



图 1-2 链接 JavaScript 代码

1.3 编写 JavaScript 的工具

对于 JavaScript 脚本的编写，可以采用两种方式，一种是使用纯文本编辑器，另一种是使用现在的专业化脚本编辑工具。

1.3.1 使用纯文本编辑器

使用纯文本编辑器（如 Windows 的记事本）来编写脚本，是早期编程人员常用的一种方法。这种方法的优点是简单、易用；缺点是由于这种编辑器的主要用途就是编辑纯文本，

不具备对 JavaScript 语言的特性支持，因此，它只适用于脚本的少量编写和修改，而要进行大量的脚本编写和设计时，则需要专业化的脚本开发工具。

1.3.2 使用专业化脚本编辑工具

使用可视化工具，如 Frontpage、Dreamweaver 以及 Flash 等工具可以十分容易地在 Web 页面中加入脚本来完成一些功能。这些工具是处理 JavaScript 的专业化开发工具，具有许多处理 JavaScript 特性的功能，如代码自动生成、语法敏感编辑、调试等，因此现在的开发人员经常使用这些工具进行 Web 程序的开发，以提高效率。必须注意的是，这些工具在自动生成有关 JavaScript 代码时会加入一些冗余的代码，但这不会妨碍熟练的脚本编程人员对 Web 页中脚本的控制。

1.3.3 使用 Microsoft 脚本编辑器

Microsoft 脚本编辑器是 FrontPage 提供的一种专业化 Web 编程工具，用于在 HTML 文档和 ASP 文件内查看和编辑 HTML 标签。还可以在文件中添加诸如 JavaScript 或 VBScript 之类的脚本，并调试该脚本。

Microsoft 脚本编辑器是 Microsoft Office 的可选组件之一，如果在安装 Office 时没有指定要安装这个组件，就不能看到它。一个简单的安装方法是安装 Office 2000 或以上版本时选择“完整安装”，以后就可以随时使用它了。

1. 启动 Microsoft 脚本编辑器

要启动 Microsoft 脚本编辑器，执行以下步骤：

(1) 启动 FrontPage，并打开或新建任意一个 HTML 文件。

(2) 将工作视图切换到“网页”视图，并选择“HTML”选项卡，此时在工作区看到的是当前所编辑页面的 HTML 代码。

(3) 从菜单栏中的“工具”菜单选择“宏”，然后从弹出的子菜单中选择“Microsoft 脚本编辑器”命令，此时就会打开如图 1-3 所示的 Microsoft 脚本编辑器。

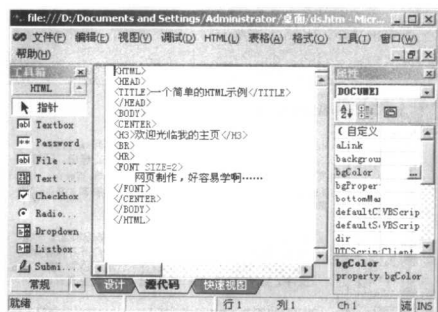


图 1-3 Microsoft 脚本编辑器

注意：如果是没有完整安装 Office，FrontPage 就会提示安装“Microsoft 脚本编辑器”。

在“HTML”视图中，脚本编辑器基于文件中文本的功能对文本进行着色。例如，用棕色表示 HTML 标签，用红色表示标签属性，用蓝色表示属性值。

2. 在 HTML 文档中插入<SCRIPT>块

使用 Microsoft 脚本编辑器可以很容易地加入<SCRIPT>块，步骤如下：

先将工作区的插入点移至整个工作区的开始位置，即<HTML>标记之前，然后在右边的“属性”窗口中将 DOCUMENT 对象的 defaultClientScript 的属性设置为“JavaScript”，此时在<HEAD>部分将出现以下代码：

```
<META name=VI60_defaultClientScript content=JavaScript>
这行代码指定这个页面的默认脚本语言是 JavaScript。
```

将工作区的插入点移至想插入<SCRIPT>块的位置（例如，刚好在</HEAD>标记之前的位置）；然后右击，此时将出现一个快捷菜单，依次选择“插入脚本块”和“客户端”命令，Microsoft 脚本编辑器将自动在当前插入点位置生成<SCRIPT>块标记，如下所示：

```
<SCRIPT LANGUAGE=JavaScript>
<!--
//-->
</SCRIPT>
```

在生成的代码中，<!-- ……//-->是 HTML 注释，而//是 JavaScript 的注释。使用这种注释标记的作用在于使较旧的不支持脚本的浏览器也能正确显示该文档的其他部分。

在<!-- ……//-->之间的空行中可以编写任意的 JavaScript 语句，如：

```
<SCRIPT LANGUAGE=JavaScript>
<!--
document.write("hello world");
//-->
</SCRIPT>
```

必须注意的是，当输入对象名和句点“document.”时，Microsoft 脚本编辑器将自动弹出一个快捷菜单，提示 document 对象所拥有的方法和属性。这种效果可以使编程人员避免写出错误的方法名或属性名。

3. 使用 Microsoft 脚本编辑器的帮助系统

JavaScript 是基于对象的脚本语言，它含有许多内置对象，而每个对象又含有大量的方法和属性。对于这些信息，编程人员没有必要完全记住，他们可以随时通过 Microsoft 脚本编辑器的帮助系统查到所需信息的详细描述。

要使用 Microsoft 脚本编辑器的帮助系统，可在 Microsoft 脚本编辑器中选择“帮助”菜单的“Microsoft 脚本编辑器的帮助”命令，这时将出现一个窗口，如图 1-4 所示。

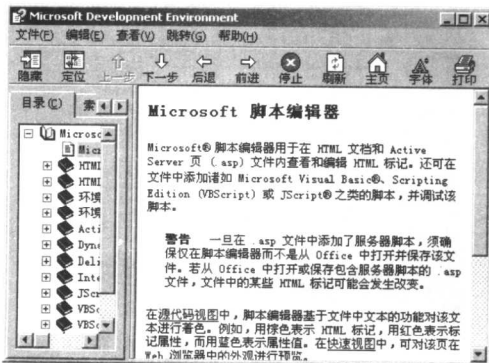


图 1-4 Microsoft 脚本编辑器的帮助系统

Microsoft 脚本编辑器的帮助系统提供了对脚本编辑器、HTML、JScript 和 VBScript 的详细描述。如果要了解 JavaScript 的有关描述，可在“Miscrosoft Script Editor 帮助”窗口左边“目录”选项卡中通过单击来展开“JScript 帮助”项目，随着左边项目的选择，窗口右边将显示左部所选择项目的详细描述。