

指南

王萍萍 裴红义 孙 强 编著

JavaScript 应用开发技术详解

从零学起，结合实践，深刻掌握JavaScript脚本编程和网页制作技术

技术要点：

- HTML语法
- JavaScript语法
- JavaScript面向对象编程
- 程序调试技术及工具
- CSS应用
- 浏览器的对象模型（Window对象、Frame对象）
- Document对象、Form对象及Cookie应用
- XML及DOM对象模型在JavaScript中的编程应用
- ASP服务器端开发技术
- Web 2.0热点——Ajax



光盘内容

书中所有范例源代码



科学出版社

北京科海电子出版社
www.khp.com.cn

TP312/2961D

2008

JavaScript 应用开发技术详解

王萍萍 裴红义 孙 强 编著

科学出版社

北京科海电子出版社

内 容 提 要

本书是一本全面介绍 JavaScript 脚本编程和网页制作技术的教程。在介绍 JavaScript 语法及相关知识的基础上,作者对 JavaScript 的高级技术进行了详细讲解,并通过实际案例对其应用开发进行全程演示。

全书共 16 章,采用理论与实践相结合的讲解方式。主要内容包括:HTML 语法,JavaScript 基本语法,JavaScript 面向对象编程,程序调试技术及工具,CSS 应用,浏览器的对象模型(Window 对象、Frame 对象等),Document 对象,Form 对象及 Cookie 应用,XML 及 DOM 对象模型在 JavaScript 中的编程应用,ASP 服务器端开发技术,以及 Web2.0 热点——Ajax。全书最后通过 6 个适用性很强的小程序对书中所讲的知识点进行了综合应用。

本书既可以作为高等院校计算机相关专业学生的 JavaScript 教材,也可作为网站设计与制作等相关从业人员的自学教程。另外,书中的大量范例对 Web 程序员极具参考价值。

图书在版编目(CIP)数据

JavaScript 应用开发技术详解/王萍萍,裴红义,孙强编著.

—北京:科学出版社,2008

ISBN 978-7-03-021792-9

I. J… II. ①王… ②裴… ③孙… III. JAVA 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 060456 号

责任编辑:何立兵 / 责任校对:杨慧芳

责任印刷:科海 / 封面设计:林陶

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

北京市艺辉印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2008 年 7 月第一版

开本:16 开

2008 年 7 月第一次印刷

印张:36.5

印数:1-4000

字数:888 千字

定价:63.00 元(含 1CD 价格)

(如有印装质量问题,我社负责调换)

前言

JavaScript 是一种语法简单、功能强大的脚本设计语言，结合 HTML，它可以在网页中实现令人炫目的特效。而在 ASP、ASP.NET、PHP、JSP 等动态网页编程技术中都可以结合 JavaScript 实现强大而丰富的功能。

本书比较全面地讲解了 JavaScript 的基础知识和应用，坚持理论与实践相结合的原则，每个重要的知识点都配有精选的范例，帮助读者理解所学内容。这些实例具有较强的通用性和可操作性。

全书共 16 章，具体内容如下：

第 1 章 主要介绍了 JavaScript 的发展简史和主要特点，JavaScript 的开发环境和开发工具。本章内容为读者呈现了 JavaScript 的知识轮廓，作为进一步学习的前导。

第 2 章 主要介绍了 HTML 基本知识，以及 HTML 常用的标签，包括格式标签、文本标签、超链接标签、图像标签、表格标签、框架标签、表单标签等。

第 3 章 主要介绍了 JavaScript 的语法规则、数据类型、变量、表达式和运算符，并重点讲解了正则表达式，使读者能够掌握 JavaScript 的基本语法知识，为后面的高级应用奠定基础。

第 4 章 详细介绍了 JavaScript 常用的 3 种控制结构和函数的相关知识，包括函数的定义、调用、参数传递等，重点讲解了函数的嵌套、递归，最后讲解了几种常用的内置函数。

第 5 章 主要介绍了 JavaScript 与面向对象编程的紧密关系，JavaScript 中对象的创建和删除、常用内置对象的属性和方法等。

第 6 章 主要介绍了 JavaScript 的调试技术，主要包括常见的错误类型、常用的调试方法，并在此基础上讲解了一些高级调试技术。

第 7 章 主要讲解了 CSS 样式表的分类和应用、CSS 选择器的分类，重点讲解了 CSS 的各种属性，最后结合实例详细讲解了 CSS 常用的滤镜。

第 8 章 主要介绍了浏览器的对象模型，重点讲解了 Window 对象和 Frame 对象，包括它们的常见属性、方法和事件，还通过实例详细讲解了它们的应用。

第 9 章 主要介绍了 Document 对象以及它的下一级对象 links 和 images，包括对象相关的属性、方法与实际应用。

第 10 章 主要介绍了 Document 对象的一个下一级对象——Form 对象，包括它的各种实际应用。该对象在网页编程中应用相当广泛，它可以包含多种元素，如文本框、按钮、列表框等。

第 11 章 主要介绍了浏览器端 Cookie 的应用，包括 Cookie 的特点和应用原理，还通过范例讲解了它在网页中的应用。

第12章 主要对XML的语法、元素、XSL、XML数据岛、XML解析器进行了介绍，还通过范例讲解了如何应用JavaScript对XML文档进行操作。

第13章 主要介绍了DOM编程基础知识，包括DOM文档对象模型，并通过DOM对象的相关属性和方法，应用JavaScript对HTML文档和XML文档进行操作。

第14章 主要介绍了JavaScript在ASP和数据库编程方面的知识。包括ASP基础知识、ASP运行环境的配置、常用ASP内置对象的使用。重点讲解在ASP中如何操作数据库，还通过范例讲解了JavaScript结合ASP对数据库进行连接、查询等操作。

第15章 主要介绍了JavaScript技术的最新应用——Ajax。重点介绍Ajax基础知识，包括定义、实现原理以及实际应用范围等，并详细讲解了Ajax的关键技术——XMLHttpRequest对象的使用。

第16章 通过几个适用性很强的实例，对前面所讲述的JavaScript各方面知识进行了综合应用，让读者体验实际项目的开发，提高实战能力。

本书既可以作为高等院校计算机相关专业学生的JavaScript教材，也可作为网站设计与制作等相关从业人员的自学教程。另外，书中的大量范例对Web程序员极具参考价值。

由于Web技术的快速发展和不断完善，加之编者水平有限，本书内容难免有纰漏和不足，恳请各位同仁和读者批评指正。

编 者

2008年5月

目 录

第 1 章 JavaScript 语言概述	1
1.1 JavaScript 简介	1
1.1.1 什么是 JavaScript	1
1.1.2 JavaScript 的主要特点	2
1.1.3 JavaScript 的用途	3
1.2 JavaScript 开发和编写工具	7
1.3 编写第一个 JavaScript 程序	15
1.3.1 编写 JavaScript 程序	16
1.3.2 在网页中嵌入 JavaScript 脚本	16
1.3.3 如何添加注释	18
1.4 本章小结	19
第 2 章 HTML 基础	20
2.1 HTML 语言概述	20
2.1.1 HTML 标签	21
2.1.2 基本 HTML 页面	22
2.1.3 特殊字符	23
2.2 HTML 字体标签	24
2.2.1 标题标签<Hn>	24
2.2.2 字体修饰标签	26
2.2.3 font 标签	27
2.3 HTML 格式标签	28
2.3.1 分段、换行标签	28
2.3.2 <center> </center> (置中) 标签	29
2.3.3 <blockquote></blockquote> (缩进) 标签	29
2.3.4 <pre></pre> (预格式化) 标签	30
2.3.5 <hr> (水平线) 标签	30
2.3.6 定义列表标签	31
2.3.7 无序和有序列表标签	32
2.4 HTML 超链接标签	33
2.4.1 设置超链接地址	33
2.4.2 设置链接目标	34
2.4.3 显示超链接的文字注释	34
2.4.4 页面内锚点链接	34
2.4.5 链接到 E-mail 地址	35

2.5 HTML 图像标签	36
2.5.1 设置图片与文字的对齐方式	36
2.5.2 鼠标经过时, 显示图片的说明文字	37
2.5.3 设置图片的大小	38
2.5.4 设置图片的边框	38
2.6 HTML 表格标签	39
2.6.1 表格属性设置	40
2.6.2 单元格属性设置	42
2.7 HTML 框架标签	44
2.7.1 框架集	46
2.7.2 框架页面	47
2.7.3 窗口的名称和链接	48
2.7.4 内置框架<iframe>	49
2.8 HTML 表单标签	50
2.8.1 文本框	50
2.8.2 按钮	51
2.8.3 单选按钮和复选框	52
2.8.4 列表框	52
2.9 本章小结	53
第3章 JavaScript 语法结构	54
3.1 语法规则	54
3.1.1 大小写敏感	54
3.1.2 空白符和换行符	55
3.1.3 可选的分号	56
3.1.4 注释	56
3.1.5 直接量	57
3.1.6 标识符	58
3.1.7 保留字	58
3.1.8 代码书写建议	59
3.2 数据类型	60
3.2.1 数值型	60
3.2.2 字符串	61
3.2.3 布尔型	62
3.2.4 空值型	63
3.2.5 未定义值	63
3.2.6 类型转换	64
3.3 变量	67
3.3.1 变量命名	67
3.3.2 变量声明	68
3.3.3 变量赋值	69

3.3.4 变量的作用域	70
3.4 表达式与运算符	71
3.4.1 表达式与运算符概述	71
3.4.2 算术运算符	72
3.4.3 赋值运算符	75
3.4.4 比较运算符	77
3.4.5 逻辑运算符	80
3.4.6 字符串运算符	81
3.4.7 其他运算符	82
3.4.8 运算符优先级	85
3.4.9 表达式中的类型转换	87
3.5 正则表达式	88
3.5.1 正则表达式的定义	89
3.5.2 RegExp 对象	89
3.5.3 正则表达式的 String 方法	95
3.5.4 高级正则表达式	98
3.6 本章小结	103
第 4 章 流程控制和函数	104
4.1 流程控制	104
4.1.1 顺序结构	105
4.1.2 选择结构	105
4.1.3 循环结构	114
4.2 函数	123
4.2.1 函数定义和函数调用	124
4.2.2 函数参数的使用	126
4.2.3 函数返回值	129
4.2.4 函数的嵌套调用	130
4.2.5 递归函数	132
4.2.6 变量的作用域	133
4.3 JavaScript 内置函数	134
4.3.1 escape 函数	134
4.3.2 unescape 函数	135
4.3.3 eval 函数	135
4.3.4 parseInt 函数	136
4.3.5 parseFloat 函数	136
4.3.6 indexOf 函数	136
4.3.7 random 函数	137
4.3.8 valueOf 函数	137
4.4 本章小结	138

第5章 JavaScript 与面向对象编程	139
5.1 面向对象编程	139
5.1.1 对象的基本概念	139
5.1.2 对象的属性和方法	141
5.1.3 创建和删除对象	142
5.2 JavaScript 内置对象	148
5.2.1 String 对象	148
5.2.2 Math 对象	153
5.2.3 Date 对象	157
5.2.4 Array 对象	162
5.2.5 Boolean 对象	168
5.2.6 Number 对象	169
5.3 事件	172
5.3.1 事件和事件处理	172
5.3.2 事件对象	174
5.3.3 JavaScript 基本事件	176
5.4 本章小结	184
第6章 程序调试	185
6.1 错误类型	185
6.1.1 语法错误	185
6.1.2 运行时错误	188
6.1.3 逻辑错误	189
6.1.4 错误消息	190
6.2 常见调试方法	192
6.2.1 使用 alert() 方法报告错误	192
6.2.2 使用 write() 或 writeln() 方法报告错误	194
6.3 高级调试技术	195
6.3.1 使用 for...in 语句检查对象属性	195
6.3.2 Navigator 中的查看点	195
6.4 Microsoft 脚本调试器	197
6.5 本章小结	201
第7章 CSS 应用	203
7.1 CSS 简介	203
7.1.1 样式定义的格式	204
7.1.2 选择器的分类	204
7.1.3 使用样式	212
7.1.4 CSS 样式的优先级	215
7.1.5 style 对象	218
7.2 CSS 属性详解	220

7.2.1 字体属性	220
7.2.2 背景属性	227
7.2.3 位置属性 (定位属性)	232
7.2.4 布局属性	235
7.2.5 边缘属性 (边框、边距、间隙属性)	239
7.2.6 列表属性	244
7.3 CSS 滤镜	248
7.4 CSS+JavaScript 综合实例	255
7.4.1 图片特效	255
7.4.2 文字特效	256
7.4.3 时钟特效	257
7.4.4 导航特效	259
7.5 本章小结	260
第 8 章 窗口和框架	261
8.1 浏览器对象	261
8.2 Window 对象	263
8.2.1 Window 对象的属性	263
8.2.2 Window 对象的方法	267
8.2.3 Window 对象的事件	281
8.2.4 Window 对象的应用实例	282
8.3 Frame 对象	287
8.3.1 创建框架页面	288
8.3.2 框架对象的属性	289
8.3.3 框架页面之间的脚本共享	292
8.4 History 对象	294
8.4.1 History 对象的属性	295
8.4.2 History 对象的方法	296
8.5 Location 对象	297
8.5.1 Location 对象的属性	297
8.5.2 Location 对象的方法	301
8.6 Screen 对象	302
8.7 Navigator 对象	305
8.8 本章小结	306
第 9 章 Document 对象	307
9.1 Document 对象	307
9.1.1 Document 对象的属性	307
9.1.2 Document 对象的方法	315
9.1.3 Document 对象的应用实例	319
9.2 links 对象	324

9.3	anchors 对象	327
9.4	images 对象	330
9.5	本章小结	335
第 10 章	form 对象	336
10.1	form 对象的属性和方法	336
10.1.1	访问表单对象	336
10.1.2	表单对象的属性	338
10.1.3	表单对象的方法	341
10.1.4	表单对象的常用事件	342
10.2	表单对象的基本元素	343
10.2.1	单行文本框	345
10.2.2	密码框	348
10.2.3	多行文本域	348
10.2.4	按钮	351
10.2.5	单选按钮	353
10.2.6	复选框	356
10.2.7	列表框	357
10.2.8	隐藏对象	360
10.3	JavaScript 表单应用实例	361
10.3.1	JavaScript 的表单数据验证	361
10.3.2	调转菜单	368
10.3.3	简单在线测试	369
10.3.4	用户注册页面	372
10.3.5	二级下拉列表	376
10.3.6	体态测试	379
10.4	本章小结	381
第 11 章	使用 Cookie	382
11.1	Cookie 概述	382
11.1.1	Cookie 简介	382
11.1.2	Cookie 属性	383
11.2	Cookie 的保存和读取	386
11.2.1	写入 Cookie	386
11.2.2	读取 Cookie	390
11.3	Cookie 应用实例	394
11.4	本章小结	398
第 12 章	JavaScript 与 XML 应用	399
12.1	XML 概述	399
12.2	XML 基本语法	403

12.2.1 XML 文档结构	403
12.2.2 声明	404
12.2.3 根元素	404
12.2.4 元素	405
12.2.5 属性	408
12.2.6 注释	408
12.3 显示和输出 XML 数据	409
12.3.1 通过 CSS 显示 XML 数据	409
12.3.2 通过 XSL 显示 XML 数据	411
12.4 XML 数据岛	419
12.5 XML 解析器	421
12.6 JavaScript 与 XML 实例应用	422
12.7 本章小结	426
第 13 章 JavaScript 和 DOM 编程	427
13.1 DOM 文档对象模型	427
13.1.1 文档结构树	427
13.1.2 节点的属性和方法	429
13.2 访问文档中的对象	433
13.3 生成节点	436
13.4 添加节点	439
13.5 插入节点	441
13.6 删除节点	444
13.7 替换节点	446
13.8 对属性进行操作	449
13.9 本章小结	452
第 14 章 ASP 服务器端开发入门	453
14.1 ASP 概述	453
14.1.1 ASP 简介	453
14.1.2 创建 ASP 页面	457
14.2 ASP 内置对象	458
14.2.1 Request 对象	459
14.2.2 Response 对象	463
14.2.3 Cookie 应用	467
14.3 JavaScript 连接数据库	470
14.3.1 ADO 对象	470
14.3.2 使用 Connection 对象连接数据库	471
14.4 JavaScript 结合 ASP 动态显示查询结果	475
14.4.1 Recordset 对象	475
14.4.2 使用 Recordset 对象读取数据并输出	478

14.5	数据库记录的分页显示	485
14.6	插入记录	489
14.7	删除记录	491
14.8	本章小结	493
第 15 章	Ajax 基础	494
15.1	Ajax 概述	494
15.1.1	Ajax 简介	494
15.1.2	Ajax 定义	495
15.1.3	传统 Web 开发技术与 Ajax Web 开发技术的区别	495
15.1.4	Ajax 的优缺点	497
15.1.5	Ajax 的应用范围	498
15.2	使用 XMLHttpRequest 对象	504
15.2.1	XMLHttpRequest 对象概述	504
15.2.2	属性和方法	505
15.3	Ajax 处理过程	510
15.3.1	第 1 步: 初始化对象并发出 XMLHttpRequest 请求	510
15.3.2	第 2 步: 指定响应处理程序	510
15.3.3	第 3 步: 发送 HTTP 请求	511
15.3.4	第 4 步: 处理服务器响应	511
15.4	Ajax 技术实例应用	516
15.4.1	多级联动下拉菜单	516
15.4.2	树形菜单	521
15.4.3	载入菜单	525
15.4.4	简单聊天室	529
15.4.5	简单留言本	543
15.5	本章小结	547
第 16 章	JavaScript 综合应用	548
16.1	鼠标跟随	548
16.2	雪花背景	551
16.3	旋转菜单	553
16.4	表针式时钟	557
16.5	提醒功能的日历	561
16.6	简单的计算器	569
16.7	本章小结	572

JavaScript 语言概述

章前导读

计算机技术和网络技术的迅速发展为网络时代的到来铺平了道路。根据最新统计数据,截至 2007 年 6 月 30 日,世界的网民总数量近 12 亿;互联网普及率达到 17.8%。网页的形式从最初的简单文字排列到现在图文并茂,滚动的字幕、游动的动画、信息框以及不断变化的状态栏提示、动态变换的时钟信息等使网页变得更加缤纷多彩;网页也从单向传递消息逐步发展为实时的、可交互的形式,如鼠标移过时变换的图像、加载网页时自动弹出的窗口、需要填写并提交的消息框等。JavaScript 在实现上述效果中可谓功不可没,本书将帮助您全面了解 JavaScript。首先让我们认识一下 JavaScript!

重点提示

- ★ 什么是 JavaScript
- ★ JavaScript 有什么特点
- ★ JavaScript 有什么用途
- ★ JavaScript 有哪些编写工具
- ★ 如何编写和运行 JavaScript 程序

1.1 JavaScript 简介

我们都知道 HTML 语言是网页设计普遍采用的一种超文本标记语言,但 HTML 自身不能为网页提供很多动态支持,它只能创建超链接以允许用户浏览不同的 HTML 文档。HTML 也不能接受用户输入,更不能对用户请求作出反应。JavaScript 是为适应动态网页制作的需要而诞生的一种新的编程语言,它可以嵌入到 HTML 文档中使网页更加生动活泼,并具有交互性。

1.1.1 什么是 JavaScript

早年 Netscape 公司为了扩展浏览器的功能而开发了一种名为 LiveScript 的脚本语言,即 JavaScript 语言的前身。LiveScript 的句法是以 Java 为基础的,但是更加简明扼要,也易于掌握。同时,LiveScript 是一种解释型语言,而 Java 则必须经过编译。Netscape 公司将 LiveScript 嵌入到 Netscape Navigator 的 β 版本中。

1995 年 6 月发布的 Netscape Navigator 2.0 B1 版本开始支持 LiveScript。1995 年 11 月，Netscape 公司与 Sun 公司达成合作协议。Sun 公司也认识到使用一种简单的脚本语言是一个好主意，于是他们正式认可了 LiveScript。这样，LiveScript 与 Java 的联系成为正式的，而 LiveScript 也改名为我们都熟悉的 JavaScript。

接着 Microsoft 公司的浏览器 Internet Explorer 3.0 也开始支持 JavaScript 了。此后，许多公司相继宣布承认 JavaScript 为 Internet 上的开放式脚本编写标准，并且把它添加到了自己的产品中。微软公司还拥有自己开发的 JavaScript，称为 JScript。JavaScript 和 JScript 基本上是相同的，只是在一些细节上稍有出入。JavaScript 短小精悍，且在客户机上执行，大大提高了网页的浏览速度和交互能力；同时它又是专门为制作网页而量身定做的一种简单的编程语言。因此，目前流行的浏览器都支持 JavaScript，JavaScript 因而日益流行起来。

1.1.2 JavaScript 的主要特点

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，是专门用来开发 Internet 客户端和服务端的应用程序，它可以很容易地嵌入到 HTML 文件中并通过浏览器中内置的 JavaScript 解释器来执行。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。另外它还能方便地同网络客户实现交互，以及使网页变得生动等。JavaScript 的出现弥补了 HTML 语言的缺陷，它是 Java 与 HTML 折中的选择，其主要特点有如下 6 点。

1. JavaScript 是一种脚本语言

脚本 (Script) 是一种能够完成某些特殊功能的指令序列 (小程序段)。这些指令序列不像一般程序那样需要被编译才能执行，而是在程序运行过程中被逐行解释的。这就意味着脚本编写起来比程序更容易一些，通常也快一些，这是由于程序是使用更为结构化和经过编译的语言来编写的。在脚本中所使用的命令与语句集称为脚本语言。

JavaScript 是一种脚本语言，它的标识符形式上与 C、C++、Java 等语言十分类似，但它采用小程序段的方式实现编程。像其他脚本语言一样，JavaScript 也是一种解释型语言，不需要先编译，而是在程序运行过程中被逐行地解释执行，相对于 C、Java 等语言来说较容易掌握，而且它与 HTML 标记结合在一起，方便了用户的使用。

2. JavaScript 是一种基于对象的语言

JavaScript 是一种基于对象的语言，这意味着它能运用已经创建的对象，但不能派生新的对象，也就是没有面向对象程序设计语言 (如 Java 就是一种面向对象的程序设计语言) 所具有的继承、多态等特点，这使得 JavaScript 更容易学习。JavaScript 主要用来访问预定义对象 (如窗体 (windows)、框架 (frames)、文档 (documents)、表单 (forms) 以及表单元素 (form elements) 等)，操纵它们的属性、方法和关联事件。

例如，在 JavaScript 中，用户不必创建“日期”这个对象，因为该语言已有这一对象，用户可以直接使用。JavaScript 本身提供了非常丰富的内部对象供设计人员使用，因而在一般情况下，用户使用这些内部对象就可以满足绝大部分需要。

3. JavaScript 是一种事件驱动的语言

JavaScript 可以直接对用户或客户的输入作出响应，无须经过 Web 服务程序。它对用户的响应，是以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。其中，事件就是指用户执行了某种操作所产生的动作，比如单击鼠标、拖动窗口、选择菜单等都可以视为事件。事件发生后，可能会引起相应的事件响应。JavaScript 是一种基于事件驱动的语言，当事件发生时，它可对之作出响应，具体如何响应某个事件取决于相应的事件响应处理程序。

4. JavaScript 是一种简单实用的语言

JavaScript 是一种脚本编写语言，它提供了一个简易的开发过程。首先，它的基本结构形式与 C、C++、Java 语言十分类似，因而学习过 Java、C 或者具有其他编程基础的使用者可以很容易地掌握这门语言。其次，JavaScript 的变量类型是采用弱类型的，并未使用严格的数据类型，这无疑给初学者带来了便利。第三，JavaScript 最杰出之处就在于可以用很小的程序做大量的事，如创建交互式网页，控制浏览器的行为等。用户无须有高性能的电脑，也无须安装复杂的软件，仅需一个字处理软件及一个浏览器，就可以使用 JavaScript 进行程序设计了。

5. JavaScript 是一种安全的语言

JavaScript 是一种安全的语言，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。虽然这在某种意义上削弱了 JavaScript 的功能，但可以有效地防止恶意程序对系统的破坏，保证了系统的安全。

6. JavaScript 是一种跨平台的语言

对于一般的计算机程序，它们的运行与平台有关。除非使用一个仿真器来模拟 Windows 环境，否则不可能在 Macintosh 上运行一个 Windows 应用程序。JavaScript 则不依赖于具体的计算机平台（虽然有一些限制），它只依赖于解释它的浏览器，只要计算能运行支持 JavaScript 的浏览器，JavaScript 就可以被正确执行。例如，不论你使用 Macintosh 还是 Windows，或是 UNIX 版本的 Netscape Navigator，JavaScript 都可以正常运行。

另外，JavaScript 还能大大减少服务器的开销，节省 CGI（Common Gateway Interface，通用网关接口）的交互时间。这是由于 JavaScript 是一种基于客户端浏览器的语言，用户在浏览中填表、验证的交互过程只是通过浏览器对调入 HTML 文档中的 JavaScript 源代码在客户端进行解释执行来完成的，而不用频繁地访问服务器。即使是必须调用 CGI 的部分，浏览器也只将用户输入验证后的信息提交给远程的服务器，从而大大减少了服务器的开销。

正是以上的这些 JavaScript 的特点，使得它在 Web 编程领域中得到了广泛的普及和运用，具有广阔的发展前景。

1.1.3 JavaScript 的用途

作为一种脚本语言，JavaScript 在许多领域都得到了广泛的应用。在使用 JavaScript 语

言进行网页制作时, JavaScript 不但可以用于编写客户端的脚本程序, 实现在 Web 浏览器端解释并执行脚本程序; 而且还可以编写在服务器端执行的脚本程序, 以实现服务器端处理用户提交的信息并相应地更新浏览器显示的 Web 服务器程序。因此可以说它是非常有用的脚本语言。下面的内容会向你介绍 JavaScript 是如何在客户端和服务器端应用的。

1. 客户端应用

JavaScript 最典型最常用的应用就是开发客户端的 Web 应用程序, 即客户端的 JavaScript 脚本程序。

客户端的 JavaScript 脚本程序是被嵌入到 HTML 文件中或链接到 HTML 页面的。当用户使用浏览器请求 HTML 文件时, JavaScript 脚本程序与 HTML 一起被下载到客户端, 由客户端的浏览器读取 HTML 文件, 并分辨其中是否含有 JavaScript 脚本。如果有, 就解释并执行它, 并以页面方式显示出来。这个过程将产生用户看到的结果。

客户端 JavaScript 具有很多功能。在 HTML 中使用 JavaScript, 我们可以利用表单元素和超链接使网页直接对用户提交的请求作出响应; 我们也能够利用警告、提示和确认信息向用户提示所发生的错误。

例如, 编写一个 JavaScript 函数用于校验用户在一个需要填入用户名的表单中输入的信息。不需要任何的网络传输, 嵌入 HTML 页面的 JavaScript 就可以检查输入的数据, 并向输入了无效数据的用户显示出错信息或出错提示对话框。

如图 1-1 所示是用户在亿邮注册邮箱时的界面。

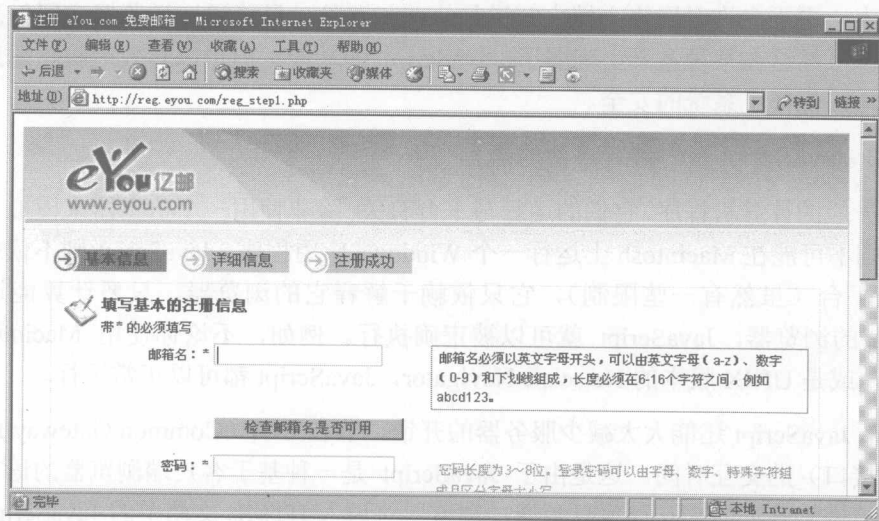


图 1-1 亿邮注册邮箱界面

当用户填写窗口中的用户名并提交后, 客户端 JavaScript 进行验证并产生适当的响应。如果用户输入的用户名不合法, 则会显示出错信息, 如图 1-2 所示。

用户可以在浏览器窗口的菜单栏中选择【查看】→【源文件】命令或在窗口空白处单击鼠标右键, 并在弹出的快捷菜单中选择【查看源文件】命令, 来查看以上网页的源文件, 我们可以看到以上网页检验用户名是否合法的功能是由 JavaScript 语言编写的, 其部分 JavaScript 代码如图 1-3 所示。