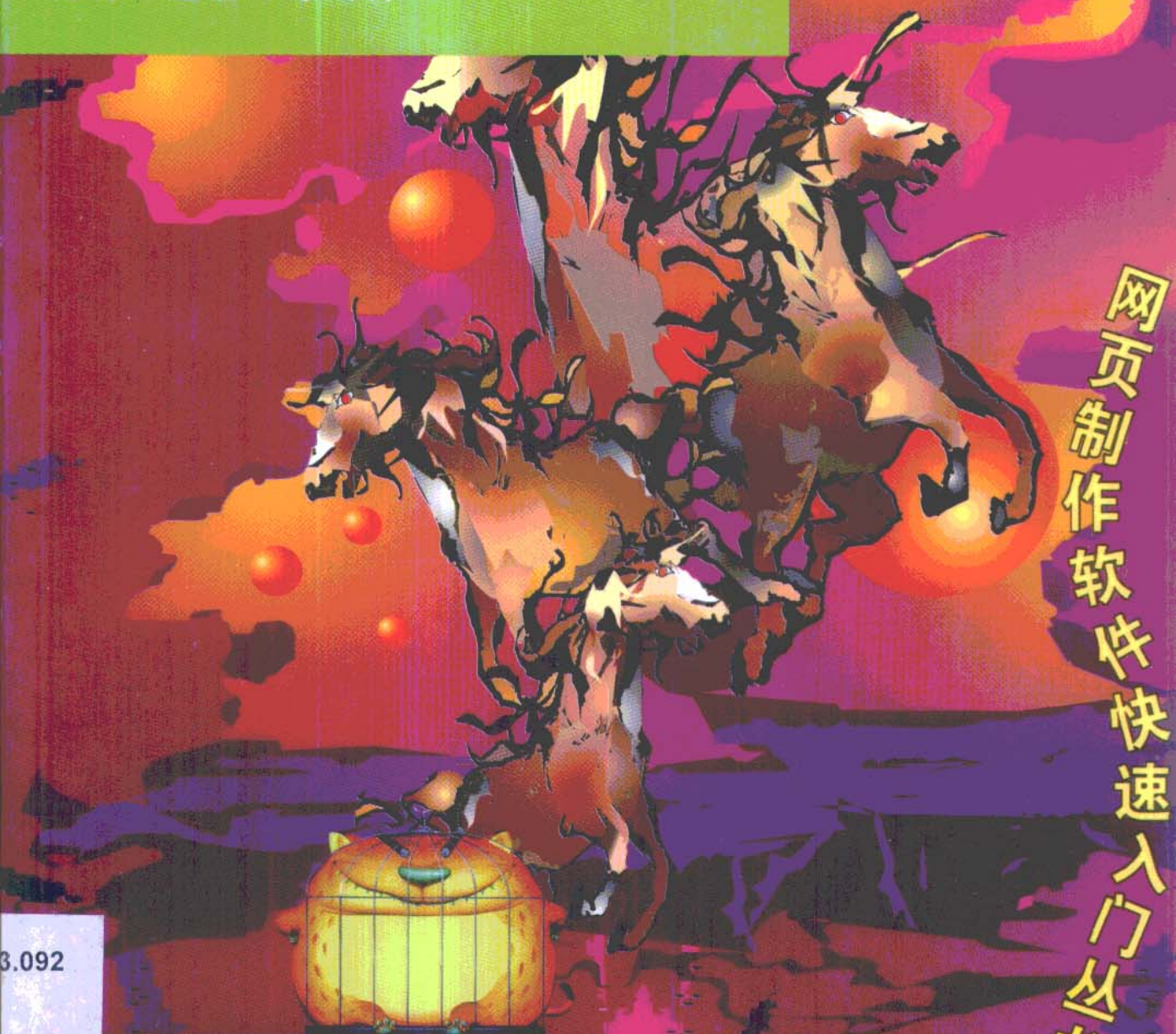
JavaScript

网页制作快速入门

杨清　施波　等编著



网页制作软件快速入门丛书

新时代出版社

网页制作软件快速入门丛书

JavaScript 网页制作快速入门

杨清　施波　等编著

新时代出版社

·北京·

图书在版编目(CIP)数据

JavaScript 网页制作快速入门/杨清等编著 .—北京：新时代出版社,2000.8
(网页制作软件快速入门丛书)
ISBN 7-5042-0532-x

Ⅰ.J... Ⅱ.杨... Ⅲ.Java 语言-主页-制作
Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2000)第 29416 号

新时代出版社出版发行
(北京市海淀区紫竹院南路 23 号)
(邮政编码 100044)
北京奥隆印刷厂印刷
新华书店经售
*
开本 787×1092 1/16 印张 16¼ 397 千字
2000 年 8 月第 1 版 2000 年 8 月北京第 1 次印刷
印数:1—4000 册 定价:22.00 元

前　言

Internet 时代，造就了新的工作和生活方式，其互联性、开放性和共享信息的模式，打破了传统信息传播方式的重重壁垒，为我们带来了新的机遇。随着计算机和信息时代的到来，人类社会前进的脚步在逐渐加快，每一天都有新的事情发生，每一天都在创造着奇迹。随着网络技术的突飞猛进，Internet 技术已经渗透到各行各业中。无论从管理方面，还是从商业角度来看，Internet 都可以带来无限生机。通过 Internet，可以实现地区、集体乃至个人的连接，从而达到一种“统一的和谐”。

近年来，我国的网站数量在不断地以惊人的速度增加，网页制作已经成为热门技术。为了使自己编辑的主页更加吸引人，网页开发者在不断地学习新技术。看到网上五花八门的主页形式，新手也在跃跃欲试。

创建交互式网页内容是许多 Web 开发者的首要目标，这可用多种办法实现。Web 浏览器通过使用视频剪辑、声音和可下载程序能够具有添加动态内容的能力。尽管其他交互式方法也能够提供吸引人的内容，但脚本语言给试图创建交互式网页的开发者提供了非常便利、有效的条件。脚本语言的最大优势在于它们易于使用。因为脚本语言直接包含在 HTML 页面中，是用纯文本创建的，创建和修改都很简单。开发者不必忍受编译和调试过程的折磨。脚本语言可以满足一个 Web 站点所要求的大多数基本交互功能，如数据验证，还可以很容易地提供交互经验，如游戏。当前主要有两种脚本语言：JavaScript 和 VBScript。

JavaScript 是一种新的描述语言，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。通过 JavaScript 可以做到响应用户的需求事件，如：form 的输入，而不用任何的网络来回传资料，所以当一位用户输入一项信息时，他不需要通过网络传送到服务器(server)端进行处理，再传回来的过程，而直接可以在客户(client)端进行事件的处理。也可以想象成有一个可执行程序在客户端上执行一样。

本书分三篇（入门篇、基础篇、提高篇），内容既简单易懂又全面、系统。各个章节都提供大量的典型范例。循序渐进，图文并茂的讲解，力图使读者尽快掌握 JavaScript 的应用技术，并熟悉它们的全貌。相信本书会成为 JavaScript 初、中级学习者很好的参考书。

网络技术发展之快令人吃惊，网络技术产品不断地推陈出新，脚本语言本身可能会相对稳定，但是新特性会不断地添加到 Netscape Navigator 和 Microsoft Internet Explorer 浏览器等相关宿主产品中。尽管在本书的写作过程中作者在不断地关注网页开发的相关技术动态，但还是存在着一定的滞后性，希望广大读者不断学习，及时掌握新技术，让你的主页与众不同！

本书由杨清、施波编写，汪钟鸣、战晓苏、仰礼友审校。参加本书编写的还有杨霞、王卫丹、

于泓、宋红、杨正辉、马军、隋利玲、王炜文。此外，周立、王峰、康建鹏、李跃华、郭伟、吴阿明、任鸿参加了资料收集和实例编写工作，董朝旭、王英、何建宇、陶涛、周利完成了录排等大量的实际工作。

对于本书的内容和编排，作者下了一番苦心，但是由于学识有限，书中偏颇和不当之处在所难免，恳请读者提出宝贵意见。

编 著 者

2000 年 3 月

第1篇 入门篇

第1章 JavaScript基本概念

在Internet时代怎样把自己的或公司的信息资源加入到WWW服务器,是广大用户日益关心的问题。采用超链接技术(超文本和超媒体技术)是实现这个目标最简单的、最快速的手段和途径。具体实现这种手段的支持环境,那就是HTML超文本标识语言,通过他们可制作所需的Web网页。

通过超文本(Hyper Text)和超媒体(Hyper Media)技术结合超链接(Hyperlink)的链接功能将各种信息组织成网络结构(Web),构成网络文档(Document),实现Internet上的"漫游"。通过HTML符号的描述就可以实现文字、表格、声音、图像、动画等多媒体信息的检索。

然而,采用这种超链接技术存在着一定的缺陷,那就是他只能提供一种静态的信息资源,缺少动态的客户端与服务器端的交互。虽然可通过CGI(Common Gateway Interface)通用网关接口实现一定的交互,但由于该方法编程较为复杂,因而在一段时间妨碍了Internet技术的发展。而JavaScript的出现,无疑为Internet网上用户带来了一线生机。可以这样说,JavaScript的出现是时代的需求,是当今的信息时代造就了JavaScript。

1.1 JavaScript简介

JavaScript的出现,使得信息和用户之间不仅只是一种显示和浏览的关系,而且实现了一种实时的、动态的、可交互式的表达功能。从而基于CGI静态的HTML页面将被可提供动态实时信息,并对客户操作进行响应的Web页面所取代。JavaScript脚本正是满足这种需求而产生的语言,它深受广泛用户的喜爱和欢迎,是众多脚本语言中较为优秀的一种,与WWW的结合有效地实现了网络计算机的蓝图。因此,尽快掌握JavaScript脚本语言编程方法是我国广大计算机用户所日益关心的。

1.1.1 JavaScript概念

JavaScript是一种新的描述语言,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。通过 JavaScript 可以做到响应用户的需求事件，如：form 的输入，而不用任何的网络来回传资料，所以当一位用户输入一项信息时，不需要通过网络传送到服务器端进行处理，再传回来的过程，而直接可以在客户端进行事件的处理。也可以想象成有一个可执行程序在客户端上执行一样。

1.1.2 JavaScript 的基本特点

JavaScript 是通过嵌入或调入在标准的 HTML 语言中实现的。它的出现弥补了 HTML 语言的缺陷，是 Java 与 HTML 折衷的选择，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，像其他脚本语言一样，JavaScript 同样已是一种解释性语言，提供了一个简易的开发过程。

它的基本结构形式与 C、C++、VB、Delphi 十分类似。但不像这些语言需要预先编译，而是在程序运行过程中被逐行地解释。它与 HTML 标识结合在一起，从而方便用户的使用操作。

2. JavaScript 是一种基于对象的语言

JavaScript 是一种基于对象的语言，同时也可以看作一种面向对象的语言，这意味着它能运用自己已经创建的对象。因此，许多功能可以来自于脚本环境中对象的方法与脚本的相互作用。

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 是一种安全性语言，它不允许访问本地的硬盘，并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，从而有效地防止数据丢失。

5. 动态性

JavaScript 是动态的，它可以直接对用户输入做出响应，无须经过 Web 服务程序，对用户的响应是采用以事件驱动的方式进行的。所谓事件驱动，就是指在主页(Home Page)中执行了某种操作所产生的动作，就称为“事件”(Event)。比如按下鼠标、移动窗口、选择菜单等都可以视为事件。当事件发生后，可能会引起相应的事件响应。

6. 跨平台性

JavaScript 只依赖于浏览器本身，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，从而实现了“编写一次，走遍天下”的梦想。

实际上 JavaScript 最杰出之处在于可以用很小的程序做大量的事。无须有高性能的电脑，仅需一个字处理软件及一个浏览器，无须 Web 服务器通道，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，可以被嵌入到 HTML 的文件之中。JavaScript 语言可以做到回应用户的需求事件(如：form 的输入)，而不用任何的网络来回传输资料，所以当一位使用者输入一项资料时，他不用经过传给服务器端处理、再传回来的过程，而直接可以

被客户端的应用程序所处理。

1.1.3 JavaScript 历史

由名称可以看出，JavaScript 是一种描述语言(而且是面向对象的)，专门用来开发 Internet 客户端和服务器端的应用程序。在客户端，通过浏览器解释嵌入在 HTML 页面中的 JavaScript 描述语句；在服务器端，用 LiveWire(JavaScript 在服务器应用方面的一部分)可以创建类似 CGI 的服务器应用程序。在客户端的 JavaScript 语句处理用户输入时不用进行网络传输，所以，可以用 JavaScript 开发快速运行的 HTML 页。

最初 Netscape 定义了 JavaScript 的原型，应用于 Netscape Navigator 上。此后 Java 语言的发明者 Sun 也支持并推广 JavaScript。在 Netscape Navigator 的不断升级中，由于 Navigator 在市场上占有优势，JavaScript 几乎成为工业标准。

JavaScript 是个新生事物，而许多人很容易把它与 Java 联系到一起，这大概是由于两者都带有"Java"这个词的缘故吧，但实际上这两者可不是一回事。其中，Java 是由 Sun Microsystem 公司开发的一种适合分布式计算的新型面向对象程序设计语言，是 C++ 的衍生语言，功能强大，但内容繁多，十分复杂，适于有一定程序编制经历或基础的人使用；而 JavaScript 却是由 Netscape 公司开发的一种脚本语言，结构简单，使用方便，对用户自身知识水平的要求并不高，易学易懂。它的代码可以直接放入 HTML 文档之中，无需编译就可在支持 JavaScript 的浏览器中运行。最初，Netscape 创建这种脚本语言时把它命名为 LiveScript，其整个语法以 Java 为基础，但比 Java 要简单扼要，同时，由于是一种脚本语言，所以无需编译，可由浏览器直接解释运行，而不像 Java 那样需要经过编译。后来，Netscape 见 LiveScript 大有发展前途，而 SUN 也觉得可以利用 Livescript 为 Java 的普及做铺垫，于是两家一拍即合，签订协议，将 LiveScript 改为 JavaScript，造就了这个强力的 Web 页开发工具。

1.1.4 JavaScript 的能与不能

JavaScript 的功能可以说是非常多的，总体可以大致归结为以下两类。

1. 交互性

使用 JavaScript 可以大大加强 Web 页的交互性，如轻松地在 Web 页中加入按钮，显示带有控制的文本，建立交互式表格等。

2. 动态性

JavaScript 可以使 Web 页上显示的文本信息动起来或是加入一些动画，从而使 Web 页看上去活泼诱人。

JavaScript 虽然强大，但是毕竟只是一种脚本语言，能力有限。以下几点是 JavaScript 所不能实现的。

1. 不能改变 HTML 编制 Web 页的一部分

使用 JavaScript 程序可以控制重新载入不同内容的页，但无法直接改变页文本的内容，即不能直接改变一张 Web 页的部分内容。在这点上，Java 语言编制的程序不受此限制。

2. 不具有通信能力

通用的 JavaScript 虽然可以制作、处理交互式表格等，但本身并不具有将数据传回服务器的能力，同时也不具有访问服务器上任何数据的能力。要想在 Web 浏览器与服务器之间进行

通信,需要使用 Java 与 CGI 语言解决。

1.2 JavaScript 与 JScript

JavaScript 是作为 LiveScript 问世的,在人们接触到它以前,Netscape 就改变了它的名字和基本原则。JavaScript 已经作了三次修改,虽然每次修改都使这个脚本编制语言在功能上更进一步,但是对于编制者来说,如果想创建能在种类尽可能多的浏览器中使用的 Web 页面,不停的改变可能是对他们的打击。令人敬畏的 JavaScript 的增强,可能无法为成千上万的访问者访问的页面带来什么好处。

微软也实现了自己的 JavaScript,被称作 JScript,现在已经是第二个主要版本了。就像 Netscape 的 JavaScript 是为 Netscape Navigator 设计的一样,JScript 是设计用在 Internet Explorer 中的。虽然这两个实现很相似,但每个都有自己独有的功能。二者之间原则上的不同,使得那些经 JavaScript 增强的精彩 Web 页面可能在"另一个"浏览器中无法使用,这可能会使成千上万潜在的用户对其敬而远之。所以,作为 Web 出版者,该怎么做呢? 要使一种脚本编制模式接受另一种? 或者试图创建 JavaScript 和 JScript 都能使用的脚本?

为了了解这些问题的答案,现将仔细研究这两种语言,其中包括如何创建在 Netscape Navigator 和 Internet Explorer 上都能运行的脚本的特别指导。

1.2.1 Netscape 的 JavaScript

当然,JavaScript 在根据实时响应把无生气的 Web 页面转化为动态的、完全交互的页面方面表现得很棒。但如果有别的要求怎么办? 如果要超越 Web 脚本编制的传统界限怎么办?

例如,如果想使 JavaScript 对于客户端和服务器端的任务都能执行怎么办? JavaScript 能够接受这样的挑战吗?

对于这些问题,可以给予响亮的回答:是。JavaScript 在一次迅雷不及掩耳的修改中,从客户端的脚本编制进化到了能满足客户端和服务器端开发需要的综合机制。使用 Web 页面时所依赖的客户端机制在 Netscape Navigator 浏览器中实现。当使用后台处理时,将依赖服务器端的 JavaScript。服务器端的机制是 Netscape 服务器的扩展,在被称为 LiveWire 的产品中实现。在 Netscape FastTrack 或者企业服务器中,LiveWire 和 JavaScript 足以满足网关脚本的需要,而且能为文件操作、数据库管理形成复杂的服务器端的解决方案。

JavaScript 是一种开放的标准,并没有把自己束缚在特定公司和产品中。当使用 JavaScript 时,就和 Netscape 的特定产品和技术紧密联系在一起了。还有,JavaScript 和 Netscape 技术模式的紧密集成会有其优势,尤其是在 Netscape 的开放网络环境(Open Networked Environment,ONE)中。

Netscape 的 ONE 是设计用作一种跨平台的开发环境,其中包括许多用来生成基于 Web 应用程序的不同技术。这个环境的关键组件是 LiveConnect,是诸如 Java、JavaScript 和插件之类的连接技术机制。使用它可以生成脚本来更新 Java 小应用程序、与 Java 小应用程序进行交互的 Netscape 插件,以及调用脚本函数的 Java 小应用程序。虽然 LiveConnect 是一个伟大的革命,但它并非是真正的独立技术,而是 JavaScript 之类现有技术的扩展机制。所以是在使用 JavaSript 的 LiveConnect 扩展来生成脚本,以使它们与 Java 小应用程序和 Netscape 插件进行交互。

1.2.2 微软的 JScript

微软几乎马上就意识到 JavaScript 的潜能，在改变脚本编制语言基础的过程中，实现了自己的相应版本——JScript。但是 JScript 对于 Web 开发来说也是一种复杂的解决方案。就像 JavaScript 紧密集成到 Netscape 技术模式中一样，JScript 也集成到了微软模式中，被设计成带有针对 Internet Explorer 浏览器的增强措施的 JavaScript 完整实现。因为这些原因，JScript 和 JavaScript 的对象机制很类似，而且可以在 Web 页面中像 JavaScript 那样使用 JScript。

JScript 和 JavaScript 在对象机制上有相似之处，但在对象模式的创建和生成方式上却有显著的不同。JScript 的核心语言功能和基准对象机制在 JScript 规范中定义，将 Internet Explorer 的对象模式针对脚本编制进行了扩展。JScript 正是在这种增强的对象模式中包含了大多数功能。将核心功能和浏览器对象模式的分离方式与 Netscape 的集成模式比较一下（在 Netscape 中核心功能被紧密集成到对象机制里），可能得出这样一个结论：微软不仅是以不同的而且是以更好的方式实现了 JScript。

Internet Explorer 对象模式和 JScript 对象模式的分离使微软能够做出一些精巧绝伦的事情，其中之一就是在浏览器中实现的其他任何脚本编制语言（例如 VBScript）可以访问 Internet Explorer 对象模式。这样微软可以通过更新 Internet Explorer 对象模式来扩展 JScript 和 VBScript 的功能。这种对象模式同样也是微软 ActiveX 机制的一部分。

ActiveX 同 LiveConnect 一样，也是用于连接技术的机制。但与 LiveConnect 不同，ActiveX 使用一种增强了的对象链接和嵌入（OLE）接口来连接 ActiveX 控制、脚本及 Java 小程序。虽然 OLE 确实不是什么新技术，但将 OLE 用于 Intérnet 却是一个破天荒的创新，有了 JScript，创建的 Web 页面可以具备诸如虚拟现实、视频上 360°的控制、实时音频、甚至视频游戏之类的功能。

在景像的背后，JScript 提供的接口和挂钩使得所有的元素能够连接在一起。ActiveX 的广泛流行使得 JScript 占有了主场之利。毕竟，如果希望自己的脚本能够利用 ActiveX 提供的所有好处，那么必须使用 JScript。即使是 Navigator 4.0 有了有限的 ActiveX 支持后也仍然是这样。

在服务器一端，JScript 的对象模式在 ActiveX 服务器机制中定义，并且在任何微软的 ActiveX 服务器中都可以获得，其中包括 Internet 信息服务器 3.0 和其后版本。

1.2.3 JavaScript 和 JScript 的比较

JavaScript 和 JScript 的多样性和动态性都来源于与各自机制的紧密结合。因此，如果使用 Netscape 客户端和服务器，JavaScript 非常理想；而 JScript 则是使用微软客户端和服务器的必然选择。

在一个企业内部网中，可以选择所支持的客户端和服务器，而在 Web 上还没有决定谁可以访问自己的网点，正是因为这个原因，熟悉两种脚本编制语言的优劣势有助于决定何时使用这个或另一个。

当查看其核心功能时，会发现 JavaScript 和 JScript 非常相似，特别是它们的语法和运算符集合。

JavaScirpt 在内置函数和控制流方面较之 JScript 要略胜一筹，因为在 Netscape Navigator

4.0 的新一版 JavaScript 中添加了 select/switch case, do-while 循环和关键字标号之类的控制流。

内置函数部分介绍了由 JavaScript 和 JScript 支持的核心功能。这些函数用来执行普通操作,如赋值操作,URL 编码字符串,将字符串转化为数字值。只有 JavaScript 支持用于数据染色的函数,激活了数据染色后,就可以克服脚本访问浏览器历史列表方面的安全限制。降低这些限制可以从包含用户访问过的每一个 Web 网点的历史列表中获取实际的 URL。

数据染色允许脚本从不同域的文档中获取信息。这样运行在框架化文档中的脚本可以访问链接图像和被装入另一个框架中的不同文档的对象。因为数据染色是一种安全考虑,所以被设计用在企业内部网或安全环境中,并且当编程者设置了特定环境变量时可以被访问。

控制流考虑了在脚本中管理程序流的方式。在这个领域,JavaScript 较之 JScript 有明显的过人之处。但记住,case-switch 和 do-while 结构可以由 JScript 使用基础控制来实现。

JavaScript 和 JScript 都是基于对象的,通过查看其对象模式可以对其有更多的了解。虽然这些对象模式原先被设计成兼容的,但两种语言还是按不同的方向发展了。结果是,两者各自都支持一个略为不同的对象集。

目前有超过 6000 万的人在访问 Web,而其中有 30% ~ 35%使用 Internet Explorer,55% ~ 60%使用 Netscape Navigator。因为这两种浏览器统治了市场,所以目的就是创建在两者中都能运行的脚本。达到这一目的的秘诀是利用 JScript 和 JavaScript 对象模式的公共部分。但是,没有几个 Web 出版者真正理解 JScript 和 JavaScript 之间的区别,而能实际指出 JScript 和 JavaScript 关系的就更少了。

开发脚本时,记住 JScript 和 JavaScript 的技术规范是可变的。随着 Internet Explorer 或 Netscape Navigator 的新版本出现,相应的语言实现就可能获得新的功能。如果希望使用其中某个所独有的功能,那么所生成的脚本就应该能在两个浏览器中都无错地运行,或者通知用户该脚本使用了在他们的浏览器中不被支持的功能。

要想使生成的脚本能够使用某个语言所独有的功能,而在每个浏览器中又都能运行,其诀窍在于提供不同的函数集,一个由 Internet Explorer 执行,另一个供 Netscape Navigator 使用。因为浏览器在执行前不检验函数,所以单独的函数不会使浏览器报错。当检查脚本时,注意用来获取浏览器名称的 navigator.appName 属性的使用,以及指定函数如何被不同的浏览器调用。

也可以使用 navigator.appName 和 navigator.appVersion 属性检验浏览器的名称和版权信息,以执行不同浏览器所独有的功能,这样在调用指定函数前先检查浏览器的名字和版本,但相应编码可能非常麻烦。使脚本合理化的另一种方式是从 navigator.userAgent 属性获取浏览器名称和版本信息,这个属性报告一个浏览器信息的简要列表。通过这种方式,只要查看一个属性而不是两个。

也可以确保只有用户浏览器支持的指定 JavaScript 版本的脚本可以运行。为此,使用 SCRIPT 标签,例如:

```
SCRIPT LANGUAGE = "JavaScript"
SCRIPT LANGUAGE = "JavaScript1.1"
SCRIPT LANGUAGE = "JavaScript1.2"
```

注意:因为 Internet Explorer 不能识别 JavaScript 1.1 或 JavaScript 1.2 的值,通过这种方式,可以保证使用各种经 JavaScript 增强的浏览器的用户可以获得 Web 页面的大部分内容。

1.3 JavaScript 和 Java

JavaScript 和 Java 很相似,但他们却是完全不同的语言! Java 是由 Sun Microsystem 公司开发的一种面向对象的程序设计语言,类似于 C + +,需要多种编译器和支持文件才能运行;与 C + + 不同的是,Java 可以独立于任何操作平台,因此就能在当今 Internet 网络各种操作平台的基础上得以迅速发展,是一种比 JavaScript 复杂得多的标准程序语言。

JavaScript 则是相对容易了解的函数式语言,JavaScript 创作者可以不那么注重编程技巧,例如声明所有的变量、类和方法,不必关心诸如 public、private 或 protected 之类的费解的东西;更重要的是,它只能存在于一个 HTML 脚本中,而且只有在装入一个兼容的浏览器时才能运行,所以许多 Java 的特性在 JavaScript 中并不支持。JavaScript 分为三类:核心版、客户方和服务器方 JavaScript,如需了解更多信息,请参阅 Netscape (http://www.netscape.com)的有关 Netscape JavaScript 介绍。目前在 Internet 上已有很多写好的 JavaScript 代码可供参考。

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 特别适合于 Internet 应用程序开发;JavaScript 则是为了扩展 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,它的前身是 LiveScript;而 Java 的前身是 Oak 语言。下面对两种语言间的异同进行比较。

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,可以用来制作与网络无关的、与用户交互作用的复杂软件,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,因而客户端必须具有相应平台上的仿真器或解释器,他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编译,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 x;
String y;
x = 1234;
y = 4321;
```

其中 x = 1234 说明是一个整数,y = 4321 说明是一个字符串。

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:

```
x = 1234;
y = "4321";
```

前者说明 x 为数值型变量,而后者说明 y 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 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 > . . . < /Script > 来标识,而 Java 使用 < applet > . . . < /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。

1.4 JavaScript 程序运行环境

1.4.1 软件环境

- Windows 95/98 或 Windows NT。
- Netscape Navigator x.0 或 Internet Explorer x.0。
- 用于编辑 HTML 文档的字符编辑器(WS、WPS、Notepad、WordPad 等)或 HTML 文档编辑器。

1.4.2 硬件配置

首先必须具备运行 Windows 95/98 或 Windows NT 的基本硬件配置环境。推荐:

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

1.5 NS 和 IE 的区别

1.5.1 相同的地方

两者都支持静态 CSS。所谓静态 CSS 就是只能读不能改。虽然两者支持的程度不一样,

但大部分基本的 CSS 性质都得到支持。

两者都支持 CSS-P,即绝对和相对定位。

两者都支持动态定位。动态定位包含物体位置的改动,物体前后词序的改动,隐蔽和显示,剪接区域(clipping area)。

两者都支持动态 HTML 内容变化。

如果用的是 NS,可以用 layer 的 document.write 来更换 layer 里的内容,比如:

```
document.layername.document.write("<H1>新内容</H1>");
document.layername.document.close();
```

如果用的是 IE,下面的语句可以帮助改变对象的内容。

```
layername.innerHTML = 新的 HTML 内容
layername.innerText = 新内容
```

两者都支持事件,比如 onClick,onMouseOver 等等。

1.5.2 不相同的地方

IE 可以改变 CSS 的性质而 NS 却不行。

IE 用 style 来改变元素的性质。如:

```
elementname.style.color = "#FF0000";
```

NS 可以装载文件到元素中,而 IE 却不能。如:

```
document.elementname.src = "external.htm";
```

IE 支持平行的 DOM,而 NS 支持树形的 DOM。

请看下面的例子来观察两者的区别。如果一个可放置的物体对象被包含在另外一个可放置的对象之内,那么 IE 可用如下的方法来读对象的性质。

```
document.all.block1.style.visibility = "hidden";
document.all.block2.style.posLeft = 20;
```

而 NS 的方法就稍有不同:

```
document.block1.visibility = "hidden";
document.block1.document.block2.left = 20;
```

第 2 章　在 HTML 中使用 JavaScript 语言

JavaScript 是一种脚本语言，可以直接嵌入 HTML 文档中，与 HTML 一起由浏览器解释执行。通过加入 JavaScript，Html 文件就变成一个在浏览器上执行的“脚本程序”，所有由 JavaScript 带来的动态效果都是由浏览器解释后产生的，可以想像有一个可执行程序在本地的浏览器上执行！但请记住 JavaScript 绝对是安全的。

2.1　JavaScript 如何写入 HTML

将脚本嵌入 HTML 文档的方法很简单，最通用的就是使用 SCRIPT 标记，即在标准的 HTML 文件中加入 SCRIPT 标记。如：

```
< Script Language = "JavaScript" > ...
< /Script >
```

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，在描述页面的 HTML 相关区域出现。

JavaScript 代码由 < Script Language = "JavaScript" > ... < /Script > 说明。在标识 < Script Language = "JavaScript" > ... < /Script > 之间就可加入 JavaScript 脚本。

为了代码不出现在不支持 JavaScript 的浏览器中，可以使用 < ! - 和 - - > 作为注释，在注释中书写JavaScript 的代码。若要在 JavaScript 的代码中使用注释，可以像 C + + 一样使用 //。支持 JavaScript 的浏览器会自动解释 JavaScript 的代码。另外，还可在/ * * /间加入注释。

JavaScript 以 < /Script > 标签结束。

在标记 < script > 中可以指定语言，如：< script language = "JavaScript" > 。在没有指定的情况下，IE 和 Netscape Navigator 默认为 JavaScript(在 IE 中还可以用 VBScript，必须指定 language = "VBScript")。

此外，还有一个标记是 < noscript > ，在不支持 JavaScript 时，浏览器显示 < noscript > 和 < /noscript > 中的内容。通常是这样使用的：

```
< noscript >
在不支持 JavaScript 的浏览器中显示的内容
< /noscript >
```

一段 JavaScript 代码可以放置于 HTML 中的任意部位,但大多数情况下习惯将其放于 </title> 标签及 </head> 标签中,因为一些代码可能需要在页面装载时就开始运行。但也可以放在描述页面的 HTML 相关区域,甚至可以在 HTML 外部装入一个 JavaScript 程序。如:

```
<SCRIPT LANGUAGE = "JavaScript" SRC = "url" >
.....
</SCRIPT >
```

这里,url 是一个外部的 JavaScript 程序,在 Netscape 承认以后缀名为 .js 的程序,而 IE 对这个要求就比较宽松,只要它符合 MIME 格式就行了。这样,如果有很多的页面需要该段程序,只需编写一个外部程序就可以在多个页面中进行调用,很方便。

JavaScript 包含下列几个最基本的概念:

(1)变量:一个代表某个值的名字就是变量。

(2)函数:具有一定功能的程序段 .JavaScript 包含 3 个内置的函数

- parseInt 可以将字符串转化为整型数。
- parseFloat 将字符串转化为附点数。
- eval 用于计算 JavaScript 表达式的值。

除了 JavaScript 内置的函数外,也可以通过关键字"function"来定义自己的函数。

(3)表达式:A = 1 + B;这就是一个表达式,它表达了一个或数个变量或数字之间的关系。

(4)语句:语句是一条由计算机完成的、达到某种目的的指令,一般说来,JavaScript 中的每一行都可认为是一条语句。

范例 2.1 和范例 2.2 就是利用 JavaScript 显示出一串文字和弹出一个具有 OK 对话框并显示()中的字符串至 Html 文件中的使用方法,其程序清单和效果图(图 2.1)如下。

范例 2.1

程序清单

```
<Html>
<Head>
<title>我的第一个 JavaScript 网页! </title>
</Head>
<Body>
用普通 html 输出的文字!
<p>
  <Script Language = "JavaScript" >
    document.write("用 JavaScript 输出的字!")
  </Script>
</Body>
</Html>
```

效果图

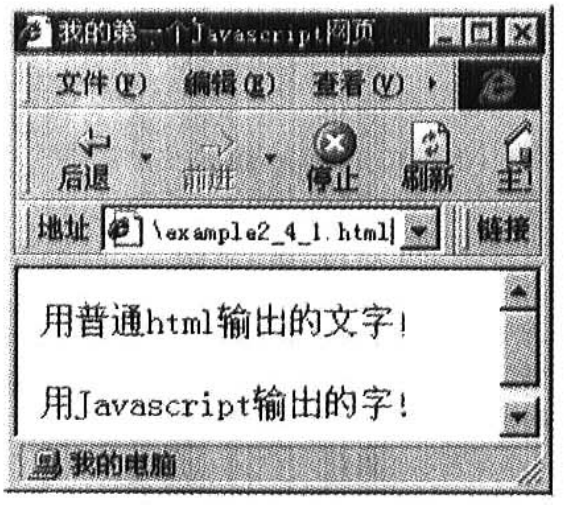


图 2.1 用 JavaScript 显示出一串文字至 html 文件中

在 Internet Explorer 4.0 中打开范例 2.1.html 文件,可以看到如图 2.1 所示的浏览窗口。

范例 2.2 利用 JavaScript 弹出一个具有 OK 对话框并显示()中的字符串至 html 文件中。其程序清单和效果图(图 2.2～图 2.5)如下。

范例 2.2

程序清单

```
<html>
<head>
<Script Language ="JavaScript">
<! - -
  alert("这是第一个 JavaScript 例子!");
  alert("欢迎你!");
  alert("JavaScript 的世界的精彩的!");
//- - >
</Script>
</Head>
<body> Hello!
<noscript>
</noscript>
</body>
</Html>
```

效果图(见下页)