



《电脑报》跟我学系列丛书

跟我学

JavaScript 编程



主编：陈宗周

编著：李晓华

云南大学出版社

《电脑报》跟我学系列丛书

跟我学 JavaScript 编程

陈宗周 主编
李晓华 编著

云南大学出版社

内 容 简 介

本书系统介绍了 JavaScript 脚本编程语言,内容主要包括 JavaScript 脚本语言概述、JavaScript 编程基础知识、JavaScript 中的控制语句和函数、JavaScript 的基本对象、JavaScript 中的事件驱动和事件处理、Navigator 环境中的对象层次以及在 JavaScript 中建立复杂交互的方法等。全书内容讲述深入浅出,便于读者循序渐进地学习。语言通俗,实例丰富,有助于读者轻松地把握 JavaScript 编程的一般方法和技巧。

本书适用于 Internet 网上编程用户和广大 Web 页面设计人员,也可作为网络爱好者的自学参考书。

跟我学 JavaScript 编程

编 者:李晓华

责任编辑:张世鸾 李继毛 余 飞

封面设计:李光宇

出 版:云南大学出版社

排 版:电脑报社照排部

印 刷:重庆日报社印刷厂

经 销:全国各地新华书店

开 本:787×1092 1/16

印 张:8.75 字 数:200 千字

版 次:1997 年 11 月第一版第一次印刷

书 号:ISBN 7-81025-853-2/TP·55

定价:10.00 元

序

在人类即将跨入 21 世纪的今天,计算机技术以前所未有的速度在全世界普及,这是全球范围的科技新潮。50 年前,当第一台计算机 ENIAC 诞生时,被称为“计算机之父”的数学家冯·诺依曼曾预测:“全世界有四台像 ENIAC 这样的计算机就足够用了”。但事实上,目前全世界的计算机已超过 4 亿台,而且还以每年新增 1 亿台的速度在发展。著名科学家在预测上出现如此大的失误,正说明计算机技术的发展远远超过人们的预料。现在计算机已经渗透进人类社会生活的各个领域,发挥着越来越重要的影响。

计算机在全球的普及,反过来对人类提出了新的要求。学习和掌握计算机知识,成了现代人类的新的需求。计算机已经象自然语言和数学一样,成了人类第三种必须掌握的工具。在今天的社会生活中,不懂计算机知识的人已经被称为“新文盲”。

全球计算机普及的热潮很自然地席卷了中国。随着中国经济的腾飞,全国广大群众渴望学习计算机知识、掌握计算机技术,这是中国正在发生的深刻变化。

要学习和掌握一门技术,需要有一个好的向导——计算机普及读物。致力于在中国普及计算机知识的《电脑报》,针对群众学习计算机知识的热潮,组织了计算机教育界的专家,精心撰写了这套适合广大群众自学计算机知识、掌握计算机技术的丛书——“跟我学”电脑系列丛书。

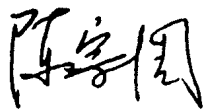
高质量的计算机普及读物应在实用性和通俗性上胜人一筹。它应该使复杂的问题变简单,使高深的专业术语通俗化。这样才能为广大群众接受。我们的计算机科普作家们在这套丛书中正是遵循这一原则,以简单、清晰的语言向广大计算机初学者及初、中级应用人员介绍计算机知识。

从内容选取上,丛书的作者也下了很大功夫。这套丛书内容涵盖面广,凡是 PC 机使用者在 90 年代中期可能遇到的软、硬件知识,都精心地编入丛书。大大增强了这套普及读物的实用性,使它既是入门向导,又是入门后的手册和参考资料。

计算机科学是实践性很强的学科,特别注重动手实践。这套书在内容中贯穿了计多操作实例,引导读者由浅入深、循序渐进地掌握对计算机的使用。希望读者勤于上机实践,加深对每一知识点的理解,从而提高计算机应用水平。

计算机充满乐趣,当你亲自动手掌握了这门技术时,你会感到极大的满足和欢乐。

愿这套“跟我学”丛书带你走进计算机世界!



1997 年 9 月

关于本书

欢迎您使用“跟我学”电脑系列丛书的《跟我学 JavaScript 编程》。

自从 WWW 作为一般的超文本系统在 Internet 上实现信息共享以来,它已成为人们用以交换信息的一种新型媒介。人们可通过它实现文本、图像、声音、数据等资源共享,同时,越来越多的人也试图提高他们的 Web 页面的交互功能。在这种情况下,Netscape 公司适时地开发出了基于其产品 Navigator 的内部脚本编程语言——JavaScript 语言。利用 JavaScript 编写 Web 页面大大提高了其交互性,从而推动了 WWW 的迅速发展。与 Internet 及 WWW 相关的众多工具一样,JavaScript 也是一种处于不断更新不断完善的编程工具。正如我们上面提及的那样,JavaScript 的应用前景是不言自明的。

本书旨在帮助读者系统而全面地学习 JavaScript 编程知识。只要读者具备 Web 页面制作的简单知识,即可进入本书内容的学习。

· 本书的结构

全书以通俗易懂的语言和大量的实例,系统而全面地介绍了 JavaScript 语言的语法、结构、命令格式以及编写脚本语言的有关知识。本书的结构如下:

第一章主要介绍了 JavaScript 脚本语言的大体情况,包括其特征、运行环境以及与其它语言的异同;

第二章通过对 JavaScript 与 HTML 的分析,让您真正走进 JavaScript 编程世界。在这一章中您会看到 JavaScript 的基本程序框架结构、基本命令语法、简单的输入输出格式、数据类型、表达式与运算符等;

第三章主要介绍了 JavaScript 的程序控制语句,还讲述了 JavaScript 的函数功能及其应用;

第四章是关于 JavaScript 中的基本对象的介绍,主要涉及对象、属性、方法、数组等内容。JavaScript 的最大特点在于它具有处理对象或创建对象的能力,因此这一章是学习 JavaScript 编程的关键部分;

第五章介绍了 JavaScript 的事件驱动及事件处理;

第六章提及了 NetScape 环境提供的各层次对象,借助这些对象更容易实现 Web 中的交互;

第七章介绍了在 JavaScript 中建立复杂交互的一些常用手段和技巧,这对于实现 Web 页面上的信息交互相当有用。

在本书的附录部分,还列举出了 JavaScript 的常用对象和运算符,以便于读者查阅有关信息。

• 本书使用的图形符号

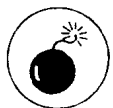
为便于您更好地学习本书,书中使用以下三种图形符号,给予您一些有益的提示:



这部分内容较深,你可以跳过去不读



这是对你很有用的经验和小技巧



注意,这是容易出错的地方

目 录

第一章 JavaScript 脚本语概况.....	(1)
1.1 JavaScript 的产生.....	(2)
1.2 JavaScript 基本特征.....	(2)
1.3 JavaScript 与 Java 的异同	(3)
1.4 JavaScript 程序开发与运行环境	(4)
1.4.1 JavaScript 运行环境	(4)
1.4.2 一个简单的 JavaScript 程序	(5)
1.5 Netscape Navigator Gold 3.0 简介	(6)
1.5.1 Navigator Gold 3.0 集成环境	(6)
1.5.2 利用 Gold 3.0 开发 Web 文档	(7)
1.5.3 怎样开发 JavaScript 应用程序	(9)
第二章 JavaScript 与 HTML 的结合	(13)
2.1 JavaScript 与 HTML 的结合	(13)
2.1.1 在 HTML 文档中加入 JavaScript 脚本	(14)
2.1.2 HTML 及 JavaScript 中的注释	(15)
2.2 JavaScript 的基本程序结构框架和命令语法	(16)
2.2.1 基本程序结构框架	(16)
2.2.2 JavaScript 中基本命令语法	(17)
2.3 最简单的输入、输出	(18)
2.3.1 JavaScript 信息的输入	(18)
2.3.2 实现信息的输出显示	(18)
2.3.3 利用输入、输出方法实现交互.....	(20)
2.4 JavaScript 中的基本数据类型	(22)
2.4.1 基本数据类型	(22)
2.4.2 常量	(22)
2.4.3 变量	(23)
2.5 JavaScript 中的表达式和运算符	(26)
2.5.1 赋值表达式	(26)
2.5.2 运算符	(27)
2.5.3 运算符的优先级	(29)

第三章 JavaScript 的基本语句和函数..... (31)

3.1 程序控制语句	(31)
3.1.1 if 条件语句	(31)
3.1.2 for 循环语句	(36)
3.1.3 While 循环语句.....	(37)
3.1.4 break 和 continue 语句	(38)
3.2 JavaScript 中的函数	(39)
3.2.1 JavaScript 中函数的定义	(39)
3.2.2 函数的返回结果	(40)
3.2.3 在函数中使用 Var 声明变量	(41)
3.2.4 函数的位置及其调用	(41)
3.2.5 函数的递归	(43)

第四章 JavaScript 中的基本对象 (46) |

4.1 有关对象的基础知识	(46)
4.1.1 对象的基本结构	(46)
4.1.2 引用对象的途径	(47)
4.1.3 对象的基本操作语句	(48)
4.1.4 对象属性的引用	(50)
4.1.5 对象方法的引用	(51)
4.2 JavaScript 中的内部对象	(51)
4.2.1 内部对象	(51)
4.2.2 JavaScript 中的内部方法	(55)
4.3 内部对象及内部方法的应用	(55)
4.4 创建新对象	(57)
4.4.1 对象的定义	(57)
4.4.2 创建对象实例	(58)
4.4.3 对象方法的使用	(58)
4.5 JavaScript 中的数组	(60)
4.5.1 使用 new 创建数组	(60)
4.5.2 创建多维数组	(61)
4.5.3 内部数组	(62)

第五章 JavaScript 事件驱动及事件处理 (64) |

5.1 什么是事件	(64)
5.2 事件处理程序	(65)
5.3 事件驱动	(65)

5.4 JavaScript 事件与对象元素	(68)
第六章 Navigator 环境中的对象层次	(70)
6.1 浏览器对象结构	(70)
6.2 文档对象	(71)
6.2.1 document 中三个主要的对象	(72)
6.2.2 document 对象中的 attribute 属性	(75)
6.3 window 对象	(75)
6.3.1 窗口对象的方法	(76)
6.3.2 窗口对象中的属性	(81)
6.4 JavaScript 输出流	(81)
6.5 历史和位置对象	(82)
6.5.1 location 位置对象	(83)
6.5.2 history 对象	(83)
第七章 在 JavaScript 中建立复杂交互	(84)
7.1 关于窗口、窗体、框架和元素	(84)
7.1.1 窗口(window)	(84)
7.1.2 窗体(form)	(85)
7.1.3 框架(frames)	(85)
7.1.4 元素(element)	(86)
7.1.5 怎样访问框架、窗体和元素	(86)
7.2 窗体对象	(87)
7.2.1 窗体对象的属性	(87)
7.2.2 窗体对象的方法	(89)
7.2.3 访问窗体对象	(89)
7.3 使用窗体元素	(89)
7.3.1 button 元素	(90)
7.3.2 checkbox 元素	(91)
7.3.3 radio 元素	(92)
7.3.4 hidden 元素	(93)
7.3.5 password 元素	(94)
7.3.6 submit 提交元素	(95)
7.3.7 reset 复位元素	(96)
7.3.8 select 选择元素	(96)
7.3.9 text 单行单列输入元素	(100)
7.3.10 textarea 多行多列输入元素	(101)
7.4 建立交互式窗体的综合应用	(102)

5.4 JavaScript 事件与对象元素	(68)
第六章 Navigator 环境中的对象层次	(70)
6.1 浏览器对象结构	(70)
6.2 文档对象	(71)
6.2.1 document 中三个主要的对象	(72)
6.2.2 document 对象中的 attribute 属性	(75)
6.3 window 对象	(75)
6.3.1 窗口对象的方法	(76)
6.3.2 窗口对象中的属性	(81)
6.4 JavaScript 输出流	(81)
6.5 历史和位置对象	(82)
6.5.1 location 位置对象	(83)
6.5.2 history 对象	(83)
第七章 在 JavaScript 中建立复杂交互	(84)
7.1 关于窗口、窗体、框架和元素	(84)
7.1.1 窗口(window)	(84)
7.1.2 窗体(form)	(85)
7.1.3 框架(frames)	(85)
7.1.4 元素(element)	(86)
7.1.5 怎样访问框架、窗体和元素	(86)
7.2 窗体对象	(87)
7.2.1 窗体对象的属性	(87)
7.2.2 窗体对象的方法	(89)
7.2.3 访问窗体对象	(89)
7.3 使用窗体元素	(89)
7.3.1 button 元素	(90)
7.3.2 checkbox 元素	(91)
7.3.3 radio 元素	(92)
7.3.4 hidden 元素	(93)
7.3.5 password 元素	(94)
7.3.6 submit 提交元素	(95)
7.3.7 reset 复位元素	(96)
7.3.8 select 选择元素	(96)
7.3.9 text 单行单列输入元素	(100)
7.3.10 textarea 多行多列输入元素	(101)
7.4 建立交互式窗体的综合应用	(102)

第一章

JavaScript 脚本语言概况

本章内容:

- ▶ JavaScript 脚本语言的产生
- ▶ JavaScript 语言的基本特征
- ▶ JavaScript 与 Java 异同
- ▶ JavaScript 开发环境及其运行环境
- ▶ Netscape Navigator Gold 3.0 简介

朋友,从现在开始就让我们一起学习 JavaScript 脚本编程语言。或许你会问,学习 JavaScript 语言有什么样的好处?它与当今 Internet 上的 WWW 有何联系?JavaScript 与 Java、HTML 又是怎样的一种关系?JavaScript 具有什么样的基本特征?

所有这些问题都是当今最热门的话题,实际上自 Windows 95 以后,人们谈论的主要话题都是 Internet,而 WWW 正是人们在网上进行信息交互的媒介。它利用超文本和超媒体技术实现了共享 Internet 网上的文本、图像、声音等多媒体信息。除此之外,人们不仅局限于 Web 的浏览,还试图在网上放入自己本人或公司的页面,并与其它用户进行动态、实时的交互。这就为计算机开发商们提出了挑战,它们必须不断创造出一些新的工具以适应用户日益增长的需求。

Netscape 公司开发的 Netscape Navigator Web 浏览器正好满足了这些用户的要求。它的出现推动了 Internet 的发展。而实质上具有革命性的改革是 Java 的推出,它能进一步利用 Internet 的潜能,使之更接近现实。当今,除了 Netscape 公司外,其他一些知名公司也正向这方面发展,他们认识到要在 Internet 上占居主导地位,必须大力研究 Web 浏览器并使其具有识别 Java 的功能。像微软公司开发的 Internet Explorer Web 浏览器与 Netscape 公司的 Netscape Navigator Web 浏览器,其功能基本上都相差无几。

在这些公司的 Web 浏览器中都内置了 HTML 超文本标识、JavaScript 脚本语言以及支持 Java(小)应用程序等主要功能。HTML 标识语言是 WWW 页面的基础,它是实现页面的基本工作及制作环境;JavaScript 可以为 Web 页面产生具有交互的功能,从而减小 Web 制作的复杂性;而 Java 是一门真正面向对象的语言,它实现了网上的真正编程。当今,这三门紧紧与 Web 制作及交互有密切关系的语言正越来越受到人们的关注。因而学习和掌握 JavaScript 编程是日益迫切的问题。

JavaScript 编程环境简单,它不需要编译器,只要拥有 Web 浏览器就可以了,而这样的浏览器到处都可以找到。若没有则可以从网上下载:

`ftp://ftp.Netscape.com/`

目前,集 HTML 制作和 JavaScript 开发及浏览为一体的集成环境不多,但有些产品还是比较好的,像 Netscape Navigator Gold 3.0 就具有开发 HTML 和 JavaScript 功能。特别是对 HTML 的开发更是非常方便,这在后面我们将看到。

1.1 JavaScript 的产生

当今随着 Internet 技术的突飞猛进,各行各业都在加入 Internet 的行业中。而 WWW 作为网上最为精采的客户程序,已成为当前 Internet 网上最受欢迎、最为流行、最新型的信息检索工具。它利用了超文本和超媒体技术结合超链接(Hyper link)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”。而描述 WWW 网上信息资源的是 HTML 超文本标识语言,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一定的缺陷,那就是它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,有效地解决了 WWW 所存在的问题,为 Internet 网上用户带来了一线生机。

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,而是实现了一种实时的、动态的、可交互的表达能力,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,它深受广大用户的喜爱和欢迎。它是众多脚本语言中较为优秀的一种,因此,尽快了解和掌握 JavaScript 脚本语言的基本知识和基本编程方法是广大用户的迫切要求。

1.2 JavaScript 基本特征

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 超文本标识语言、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。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征:

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。其基本的结构形式与 C、C++、VB、Delphi 十分类似,但它不像这些语言一样,需要先编译,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。

JavaScript 的简单性还主要体现在: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。

2. 基于对象的语言

JavaScript 是一种基于对象的语言,同时也可以看作是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. 动态性的

JavaScript 是动态的,它可以直接对用户或客户输入做出响应,无须经过 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,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。

4. 具有较强的安全性

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“病毒”的入侵。

5. 多平台性

实际上 JavaScript 是依赖于浏览器本身,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。

1.3 JavaScript 与 Java 的异同

JavaScript 与 Java 虽然有紧密的联系,但却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,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,它的前身是 Live Script,这是与 LiveWire 服务协议有关的一门语言;而 Java 的前身是 Oak 语言。下面对两种语言间的异同作如下概括:

1. JavaScript 是一种非常有用的语言,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(Object Based)和事件驱动(Event Driven)的编程语言。

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。

2. 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。

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。

3. 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。

Java 则是一种与 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。

4. 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 $X = 125$, $X = '125'$ 前者说明 X 为数值型变量,而后者说明 X 为字符型数量。

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。 $X = 123$, $Y = 123$, 其中 $X = 123$ 说明是一个整数, $Y = 123$ 说明是一个字符串。

5. JavaScript 采用动态联编,即 JavaScript 的对象引用在运行时进行检查,如不经运行就无法实现对象引用的检查;

Java 则采用静态联编,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。

6. 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. JavaScript 提供了足够的能力来建立自身对象的方法。但它与 Java 中所提供的类和继承性是不尽一样的。因而它的扩展的功能受到了一定的限制。

Java 与其它面向对象的编程语言一样,能够真正创建自己的类,具有全部扩展的能力。

8. 虽然 JavaScript 制作的 Web 页面具有实时、动态的交互能力,但这种交互能力是有限的;它除了能分析、建立和调用 URL 外,并不能直接与 Web 服务器交谈。

Java 具备了真正与 URL 工作及 HTTP 服务器交谈的能力,特别适合分布式的 Internet 网络。

JavaScript 程序开发与运行环境

1.4.1 JavaScript 运行环境

1. 软件环境

- Windows 95 或 Windows NT。
- Netscape Navigator 2.0 以上版本或 Internet Explorer 3.0 以上版本。其中 Netscape 浏览器完全支持 JavaScript,而 Internet Explorer 3.0 只是部分支持 JavaScript。
- 用于编辑 HTML 文档的字符编辑器(Notepad、wordpad 等)。

2. 硬件配置

首先必须具备运行 Windows 95 或 Windows NT 的基本硬件配置环境。如 PC 机:

- 基本内存 8MB
- 显示器至少需要 256 色,分辨率在 640×480 以上。
- CPU 至少 486 以上。
- 鼠标和其它外部设备(根据需要选用)。

1.4.2 一个简单的 JavaScript 程序

下面我们通过一个 JavaScript 程序的开发例子,说明它的脚本是怎样被嵌入到 HTML 文档中的。程序在 Netscape 中运行后的结果见图 1-1 所示。

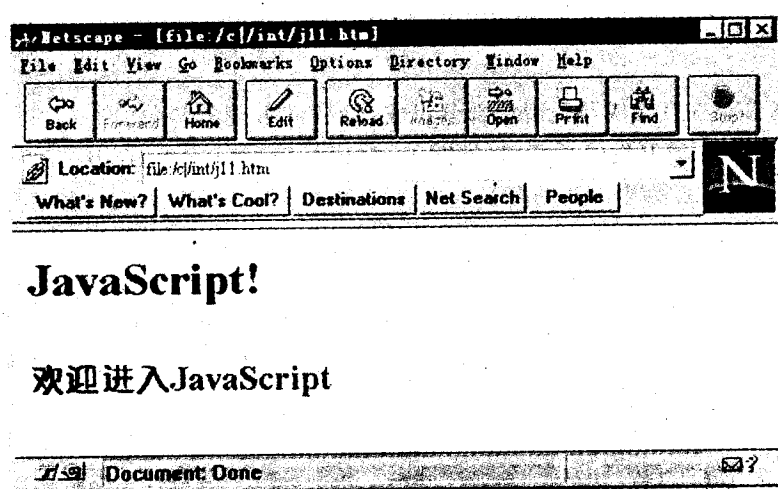


图 1-1 JavaScript 程序运行结果

源程序为:

```
< HTML >
< HEAD >
< SCRIPT LANGUAGE = " JavaScript " >
    document.writeln( ' < H1 > JavaScript! < /H1 > ' );
    document.writeln( ' < BR > ' );
    document.write( ' < H2 > 欢迎进入 JavaScript < /H2 > ' );
< /SCRIPT >
< /HEAD >
< /HTML >
```

可以看出,这个文件是 HTML 文档,其标识格式为标准的 HTML 格式。

包含在 HTML 中的 JavaScript 代码由 < SCRIPT LANGUAGE = " JavaScript " > ... < /SCRIPT > 说明。

通过上面的实例,可看到编写一个 JavaScript 程序确实非常容易。

1.5 Netscape Navigator Gold 3.0 简介

目前 World Wide Web 网上开发的支持环境较多,其中 Netscape Navigator Gold 3.0 是比较出色的,它除了能开发出所见即所得的 Web 页面以外,还增加开发 JavaScript 脚本语言的功能。虽然这种开发方式对 JavaScript 的开发能力有所限制,但仍然实现了 HTML 和 JavaScript 集成化的开发,因而选择这种开发工具是本书鼎力推崇的。

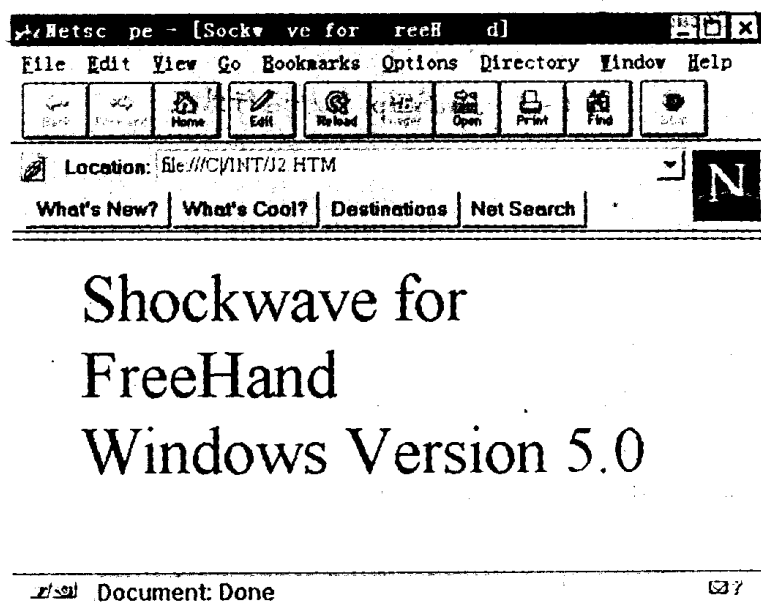
1.5.1 Navigator Gold 3.0 集成环境

Navigator Gold 3.0 集成环境由两部分组成:一是 Web 浏览器(如图 1-2(a)),它支持 HTML 3.2 标准。二是开发编辑窗口(如图 1-2(b)),该窗口支持 HTML 文档的所见即所得的开发和 JavaScript 脚本语言的开发。由这两个窗口构成强有力的 Web 制作和是否有遗漏的开发。用户可以非常方便地切换这两个窗口。

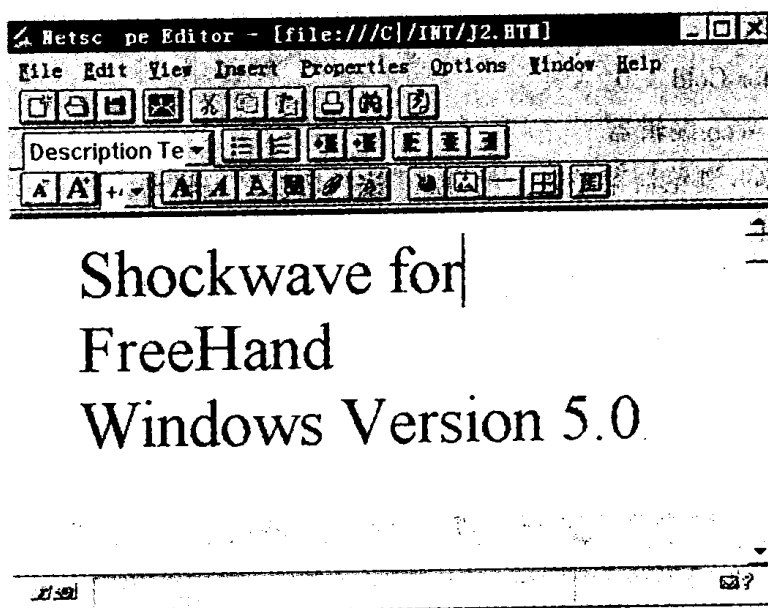
要从浏览窗口到 Web 编辑窗口,只需在浏览窗口中击“Edit”按钮即可。

若要从编辑窗口切换到浏览窗口,可选择“File”菜单的“Browse Document”命令。

在编辑器窗口中提供了 Web 页面中 HTML 文档的建立、打开、保存、浏览、裁剪、复制、粘贴、打印等有关内容。



(a) 浏览窗口



(b) 编辑窗口

图 1-2 Netscape Navigator Gold 3.0 开发集成环境

1.5.2 利用 Gold 3.0 开发 Web 文档

在这里我们所说的 Web 文档包括两个部分,这就是 HTML 和 JavaScript。

下面我们通过一个例子来说明利用 Gold 3.0 开发 Web 文档的过程。图 1-3 显示其结果。

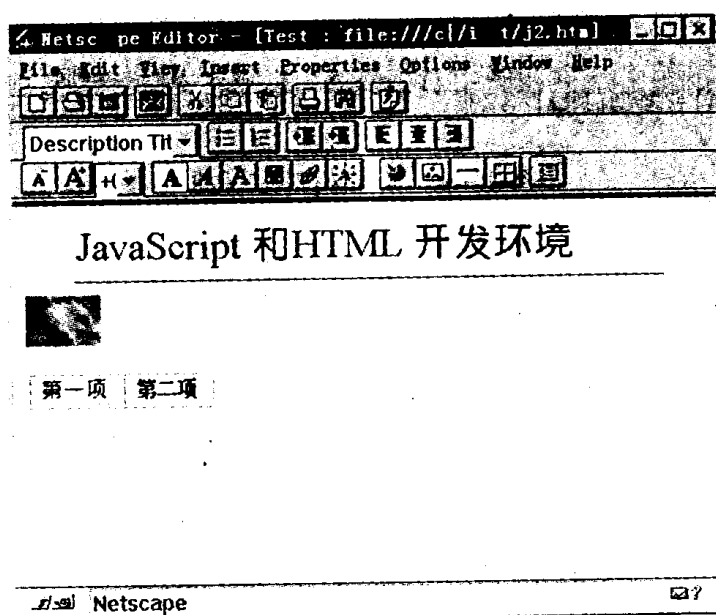


图 1-3 程序执行结果