

即学即用

JavaScript

林嵩 编著

人民邮电出版社

12

计算机实用软件丛书

即学即用 JavaScript

林 嵩 编著

人民邮电出版社

图书在版编目(CIP)数据

即学即用 JavaScript/林嵩编著. - 北京:人民邮电出版社,1998.4

(计算机实用软件丛书)ISBN 7-115-06851-8

I. 即… II. 林… III. 程序语言,JavaScript - 基本知识 IV. TP312

中国版本图书馆 CIP 数据核字(98)第 03316 号

内 容 简 介

JavaScript 是一种简洁的、面向对象的、跨平台的描述语言,是一种专门用来开发 Internet 客户端和服务端的应用程序。使用 JavaScript 可以快速创建类似 CGI(公共网关接口)程序、实现交互的动态 HOMEPAGE。因而它可以说是网络时代最重要的语言之一。

本书对 JavaScript 语言的内容、功能、特性和实际应用做了深入浅出的全面介绍,对对象、窗口和帧等的使用技巧做了详细的论述,并给出了全部 JavaScript 的对象、属性、方法的详细参考和实例,使读者能在较短的时间内掌握 JavaScript 程序设计的方法和技巧。

本书适合于想制作动态 HOMEPAGE 的网络技术人员和 Internet 爱好者,也可作为网络技术培训教材和网络软件开发人员的参考资料。

计算机实用软件丛书

即学即用 JavaScript

Jixue Jiyong JavaScript

◆ 编 著 林 嵩
责任编辑 姚予疆

◆ 人民邮电出版社出版发行 北京崇文区夕照寺街 14 号
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092 1/16
印张:17

字数:418 千字

1998 年 5 月第 1 版

印数:1-4 000 册

1998 年 5 月北京第 1 次印刷

ISBN 7-115-06851-8/TP·578

定价:22.00 元

“计算机实用软件丛书”编委会

高级顾问 张效祥 胡启恒

主任 牛田佳

副主任 李树岭 罗晓沛

特约编委 谭浩强 陈树楷

编委 (按姓氏笔画排序)

毛波 方裕 史美林 孙中臣

孙家骥 刘炳文 刘德贵 吴文虎

张国锋 周山芙 周堤基 钟玉琢

柳克俊 侯炳辉 赵桂珍 聂元铭

徐国平 徐修存 寇国华 戴国忠

丛书前言

JS086/1

随着计算机、通信和信息技术的迅速发展与广泛应用,人类正在进入信息化社会。计算机技术的应用与推广,将直接推动社会信息化的发展;而计算机技术的应用与推广,实质上取决于计算机软件的应用和推广,可以说,没有软件,就没有计算机的应用;学习、使用计算机,从根本上讲就是学习和掌握软件的使用。

为了适应当前计算机技术发展的需要,满足读者学习、使用计算机软件的需求,人民邮电出版社约请有关专家编写出版了这套“计算机实用软件丛书”。

这套丛书的特点是:普及兼顾提高,应用兼顾开发,各书独立成册形成系列,并注重其相关性,使丛书成为广大计算机应用和开发人员学习使用计算机的必备用书。

这套丛书的内容包括:程序设计语言、操作系统技术、数据库技术、软件开发技术及工具、网络技术、多媒体技术等。

在计算机技术飞速发展的今天,软件产品更新快,经常有新产品或新版本问世,因此我们不但介绍当前流行和优秀的软件,而且力求尽快把国内外最新的软件产品也介绍给读者。

我们将全心全意为读者服务,也热切期待广大读者对丛书提出宝贵意见,以进一步提高丛书的质量。让我们共同努力,为提高我国的计算机开发、应用水平做出贡献。

“计算机实用软件丛书”编委会

前 言

JavaScript 是 Netscape 公司推出的一种简洁的、面向对象的、跨平台的描述语言,其前身由 Netscape 公司定义,后由于 Java 语言的发明者 SUN 小型系统公司的支持才得以迅速发展,现在它已经成为主流浏览器 Netscape Navigator、Netscape communicator、Internet Explorer 都支持的标准 Internet 开发语言。

JavaScript 分为 Navigator JavaScript 和 LiveWire JavaScript 两类,分别用来开发 Internet 客户端和服务端的应用程序。

嵌在 HTML 页中的 Navigator JavaScript 描述语句能对诸如鼠标点击、表单输入、页面浏览等用户事件作出反应。不用任何网络传输,浏览器就能解释执行嵌入 HTML 文档中的 JavaScript 语句,进行相应的事件处理。

LiveWire JavaScript 可开发服务端的应用程序,与标准的 CGI 程序相比较,LiveWire JavaScript 被直接集成在 HTML 文档中,易于快速开发并极易维护。LiveWire JavaScript 包含一个对象构架,此对象构架能用来为支持交叉用户请求、多重用户、多重应用程序提供数据。LiveWire JavaScript 也提供数据库访问的对象和方法,这些服务的界面与 SQL(Structured Query Language)database server 的界面相似。

JavaScript 与 Java 语言既有区别又有联系,不像 Java 通过描述语句来构建类的编译时间系统,JavaScript 支持运行时间系统,这种系统由少量表示数字值、逻辑值、字符串值的数据类型构成;JavaScript 没有 Java 的静态类型和严格的类型检验;JavaScript 有一种简单的、基于引例的对象模型,它能提供很多强有力的功能;JavaScript 也支持不需要作任何特定声明的函数,函数可作为对象的属性来对待,执行起来就像没有严格限制的类型化的方法一样。JavaScript 支持大多数 Java 的表达语法和基本的流控制结构,是与 HyperTalk、dBASE 类似的短小、精悍的描述语言。因为其语法简单,容易学习使用,所以也受到了广大 Internet 用户的欢迎。

本书全面系统地由浅入深地介绍了 JavaScript 语言的各个方面,包括对 JavaScript 语言的内容、功能、特性和实际应用;对象、窗口和帧等的使用技巧;并给出了全部 JavaScript 的对象、属性、方法的详细参考和实例,使读者能在较短的时间内掌握 JavaScript 程序设计的方法和技巧。

本书适合于想制作动态 HOMEPAGE 的网络技术人员和 Internet 爱好者,也可作为网络技术培训教材和网络软件开发人员的参考资料。

在本书的编写过程中,得到了 Netscape 公司 Jennifer Penunuri 的支持,在此表示感谢。由于作者水平有限,书中疏漏和不妥之处,敬请广大读者和专家批评指正。

编著者

目
录

第一章 JavaScript 介绍	1
1.1 学习 JavaScript 从何入手	1
1.2 本书的一些约定	2
1.3 JavaScript 的含义、基本概念	2
1.3.1 什么是 JavaScript	3
1.3.2 在 HTML 中嵌入 JavaScript	4
1.3.3 定义函数和调用函数	8
1.3.4 使用 write 方法	9
1.3.5 编写事件处理语句	11
1.3.6 确认表单输入	14
第二章 使用 Navigator JavaScript 对象	17
2.1 Navigator 对象分级体系	17
2.2 一个文档属性的例子	18
2.3 JavaScript 反映值和 HTML 布局	20
2.4 关键的浏览器对象	21
2.4.1 window 和 Frame 对象	21
2.4.2 document 对象	21
2.4.3 Form 对象	22
2.4.4 location 对象	22
2.4.5 history 对象	22
2.4.6 navigator 对象	23
2.5 Navigator 的对象数组	23
第三章 窗口和帧用法	25
3.1 打开和关闭窗口	25
3.1.1 打开窗口	25
3.1.2 关闭窗口	26
3.2 使用帧	26
3.2.1 创建一个帧	26
3.2.2 更新帧	28
3.2.3 创建和更新帧的例子	28
3.3 引用窗口和帧	29
3.3.1 引用窗口的属性、方法和事件处理	30

3.3.2 在表单提交和超文本链接时对窗口的引用	31
3.4 在窗口和帧之间浏览	31
第四章 实时连接	33
4.1 实时连接准备知识	33
4.1.1 使能实时连接	33
4.1.2 Java 控制台	33
4.1.3 Netscape 压缩包	34
4.1.4 使用 Netscape 压缩包	34
4.2 JavaScript 对 Java 的数据传递	35
4.2.1 直接访问 Java	35
4.2.2 控制 Java applet	35
4.2.3 控制 Java 插件	37
4.2.4 数据类型转换	38
4.3 Java 对 JavaScript 的数据传递	38
4.3.1 获得 JavaScript 窗口的句柄	38
4.3.2 访问 JavaScript 对象和属性	39
4.3.3 调用 JavaScript 方法	39
4.3.4 数据类型转换	41
第五章 进一步的讨论	43
5.1 JavaScript URL 用法	43
5.2 客户端图像位图用法	44
5.3 服务器端图像位图用法	44
5.4 状态条用法	45
5.5 cookie 用法	45
5.5.1 一些限制	46
5.5.2 在 JavaScript 中使用 cookie	46
5.5.3 一个使用 cookie 的例子	47
5.6 数据加扰的用法	48
5.6.1 加扰操作情况	48
5.6.2 使能加扰	49
5.6.3 加扰或取消加扰单一的数据成分	49
5.6.4 条件性描述语句产生的加扰	49
5.7 判定已经安装的插件	50
5.7.1 mimeTypees 数组	50
5.7.2 plugins 数组	51
第六章 值、变量和常量	53
6.1 值	53

6.1.1 数据类型转换	53
6.2 变量	54
6.3 常量	54
第七章 表达式和运算符	57
7.1 表达式	57
7.1.1 条件表达式	57
7.2 运算符	58
7.2.1 赋值运算符	58
7.2.2 比较操作符	59
7.2.3 算术运算符	59
7.2.4 位操作符	59
7.2.5 逻辑运算符	61
7.2.6 字符串运算符	61
7.2.7 特殊运算符	61
7.2.8 运算符的优先级	63
第八章 对象模型	65
8.1 对象和属性	65
8.2 函数	66
8.2.1 定义函数	66
8.2.2 使用函数	66
8.2.3 使用参数数组	67
8.3 创建新对象	68
8.3.1 索引对象属性	69
8.3.2 为对象类型定义属性	69
8.3.3 定义方法	70
8.3.4 使用 this 作为对象参照	70
8.3.5 删除对象	71
第九章 内建的对象和函数	73
9.1 内建的对象	73
9.1.1 Array 对象	73
9.1.2 Boolean 对象	74
9.1.3 Date 对象	75
9.1.3.1 Date 对象的方法	75
9.1.3.2 使用 Date 对象的例子	76
9.1.4 Function 对象	77
9.1.5 Math 对象	78
9.1.6 Number 对象	79

9.1.7 String 对象	79
9.2 内建的函数	80
9.2.1 isNaN 函数	80
9.2.2 parseInt 和 parseFloat 函数	81
第十章 JavaScript 语句	83
10.1 条件语句	83
10.2 循环语句	84
10.2.1 for 语句	84
10.2.2 while 语句	85
10.2.3 break 语句	86
10.2.4 continue 语句	86
10.3 对象处理语句和运算符	86
10.3.1 new 运算符	87
10.3.2 this 关键字	87
10.3.3 for...in 语句	88
10.3.4 with 语句	88
10.4 注释	89
第十一章 Netscape 包	91
11.1 netscape.javascript.JSObject	91
11.2 netscape.javascript.JSException	93
11.3 netscape.plugin.Plugin	94
11.3.1 构造程序和方法	94
第十二章 Navigator JavaScript 参考	97
附录 保留字	259

第

章

JavaScript 介绍

JavaScript 是一种简洁的、面向对象的描述语言,是专门用来开发 Internet 客户端和服务端的应用程序。Netscape Navigator 作为客户端浏览器,解释嵌入 HTML 页面中的 JavaScript 描述语句;LiveWire 用于服务器端,可以创建类似 CGI(Common Gateway Interface)程序的基于服务器的应用程序。透过 JavaScript 可以做到回应用户的需求事件(如:表单的输入)而不用任何的通路来回传输资料。所以当一位使用者输入一项资料时,它不用经过传给服务器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。因此,利用 JavaScript 可以开发出快速运行的动态 HTML 页。

JavaScript 语言原型由 Netscape 公司定义,用在公司自己开发的 Netscape Navigator 浏览器上,后得到 Java 语言发明者 SUN 小型系统公司的支持,逐渐发展完善,随着 Navigator 浏览器的不断升级,内嵌的 JavaScript 语法也在不断发展变化。因 Netscape 公司的浏览器在市场上占有优势,所以 JavaScript 已经成为事实的标准。Microsoft 公司的 Internet Explorer 浏览器也对其完全兼容。

出于对市场的争夺,Netscape 和 Microsoft 公司之间展开了浏览器大战。在这两个主流浏览器的技术标准制定上,双方存在很大的分歧,如:push 技术、DHTML(动态 HTML)技术等。对于 JavaScript 语言,Microsoft 公司虽然全兼容 Netscape 公司的语法标准,但也制定了另外的一些独有 IE 浏览器支持的对象、方法等,称为 JScript。但不管浏览器之争谁是胜利者,都不会改变 JavaScript 在 Internet 应用开发中的地位。JavaScript 确实是 Internet 开发中简洁高效的语言。

1.1 学习 JavaScript 从何入手

学习 JavaScript 以前,读者应掌握一些有关 Internet(因特网)和

WWW(World Wide Web)的知识;应熟悉 HTML(HyperText Markup Language)的语法,并有一定的编制 Web 页的经验。当然,如果读者曾经有过 C 语言或 Visual Basic 的编程经历,理解掌握 JavaScript 会更快一些,但没编程经验也没关系,一样能掌握并运用它。

本书首先介绍 JavaScript 并描述它在 Navigator 中的使用情况,主要说明客户端的 JavaScript 的特性,也涉及了一些高级的应用论述。然后详细解释 JavaScript 描述语言自身的语法和符号含义,包括 JavaScript 内含的字符串、数学、日期等对象,内含的函数及描述语句语法,这些内容既可用于客户端,也可用于服务器端的程序开发。最后给出了详尽的按字母顺序排列的 JavaScript 的全部对象、属性、方法及 JavaScript 在 HTML 页中的描述语句。

如果读者以前并未接触过 JavaScript,就应该从头看起,从前几章可以学到大部分的 Navigator JavaScript 的知识,请着重阅读“对象模型”一章;如想实现特别的要求,可参考最后按字母顺序排列的 JavaScript 成分的索引。等对 JavaScript 有了一定了解后,可回过头再仔细读读“窗口和帧用法”和“进一步的讨论”两章,会有更深的理解。如果是 LiveWire 应用程序开发人员,则应仔细阅读每一部分,它们都可应用于 LiveWire JavaScript 程序的开发;如果只想把客户端功能嵌入应用程序,那么只阅读关于客户端 JavaScript 应用部分就足够了。

1.2 本书的一些约定

Netscape Navigator 可以在很多不同的操作系统上运行,下面的信息适用于所有的版本。在 Windows 中文件和目录路径表示时用反斜杠分隔目录名,对 UNIX 版本来说,同样的路径中用正斜杠代替反斜杠。

本书中的统一资源定位地址(Uniform Resource Locators)的形式如下:

`http://server.domain/path/file.html`

在这些 URLs 中,server 代表运行 LiveWire 的服务器的名字,如:“www”或“home”;domain 表示 Internet 域名,如:“netscape.com”或“microsoft.com”。如果设置了代理服务器,可能就不需要为在此域内的客户机指定域名了。一般说来,在 URLs 中的斜体项是变量,正常字体表示常量。

JavaScript 方法和函数名用黑体表示。

JavaScript 对象和属性名用斜体表示。

1.3 JavaScript 的含义、基本概念

这一节介绍 JavaScript 的含义,讨论一些 JavaScript 在 Navigator 浏览器中的基本概念,同时提供了一些应用实例。读者只要理解了这些实例,把这些实例稍作修改,嵌入自己的 HTML 代码中,就可实现一定的功能。

1.3.1 什么是 JavaScript

JavaScript 是 Netscape 公司的跨平台的、面向对象的、用来开发客户和服务器的应用程序的描述语言。有两个类型的 JavaScript:

- (1) Navigator JavaScript, 也称为客户端 JavaScript;
- (2) LiveWire JavaScript, 也称为服务器端 JavaScript。

1. Navigator JavaScript

Netscape Navigator 2.0 及其后来版本能解释嵌入 HTML 页中的 JavaScript 描述语句。当客户端浏览器显示含有 JavaScript 描述语句的页面时,服务器通过网络把包含 HTML 及 JavaScript 描述语句的整个文档传送到客户端,Navigator 显示 HTML 的同时执行 JavaScript,创建用户看到的结果。这个过程可用图 1-1 来表示。

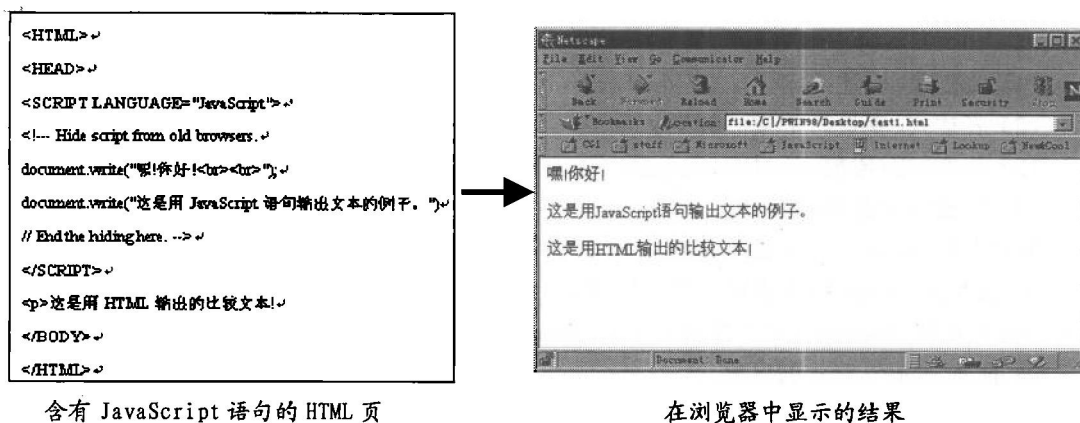


图 1-1

嵌在 HTML 页中的客户端 JavaScript 描述语句能对诸如鼠标点击、表单输入、页面浏览等用户事件作出反应。例如:可以写一个 JavaScript 函数来验证用户在一个要求输入电话号码或邮政编码的表单中输入的信息是否正确。不用任何网络传输,Navigator 浏览器解释执行嵌入 HTML 文档中的 JavaScript 语句,能检验输入的数据,假如输入不正确,会弹出对话框警示用户。

2. LiveWire JavaScript

LiveWire 是用 JavaScript 创建基于服务器的应用程序的一个应用程序开发环境。开发出的这种应用程序与 CGI(Common Gateway Interface)程序很相似。与 Navigator JavaScript 相反,LiveWire JavaScript 应用程序被编译成字节码的可执行文件。这些可执行的应用程序的执行需要有包含 LiveWire 服务器扩展的 Netscape server 2.0 或更高版本的支持。

LiveWire 服务器扩展动态地创建 HTML,这个 HTML 页(可能包含客户端 JavaScript 描述语句)接着被服务器通过网络发送到 Navigator 客户端,在客户端显示结果。这个过程可用图 1-2 表示。

与标准的 CGI 程序相比较,LiveWire JavaScript 被直接集成在 HTML 页中,易于快速开发并极易维护。LiveWire JavaScript 包含一个对象构架,此对象构架能用来为支持交叉客户请求、多重客户、多重应用程序提供数据。LiveWire JavaScript 也提供数据库访问的对象和方法,这些服务的界面与 SQL(Structured Query Language)database server 的界面相似。

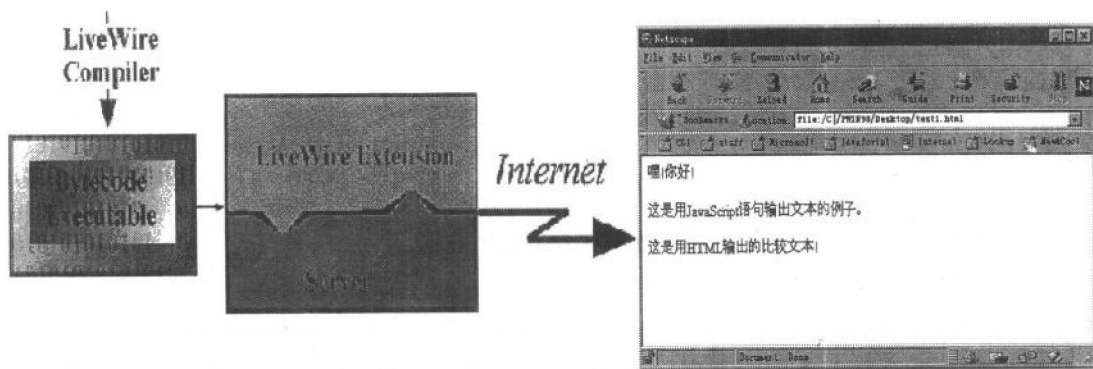


图 1-2

3. Navigator JavaScript 与 LiveWire JavaScript 的共同点

正像前面部分所描述的,客户端与服务端 JavaScript 有大量的不同之处,但它们有以下共同点:

- (1) 关键字、描述语句语法、基本语法;
- (2) 表达式、变量和常量的规则;
- (3) 基本的对象模型(尽管 Navigator 和 LiveWire 有不同的对象构架);
- (4) 内建的对象和函数。

所以,用 LiveWire 常常能写出既适用于客户端又适用于服务器端的函数。

不同的 JavaScript 版本适用于特定的 Navigator 版本的浏览器。如:JavaScript 1.1 是对应于 Navigator 3.0 的,JavaScript 1.2 是对应于 Communicator 4.X 的。

4. JavaScript 与 Java 的区别

JavaScript 与 Java 有点相似,但它们是不同的。JavaScript 没有 Java 的静态类型和严格的类型检验。JavaScript 支持大多数 Java 的表达语法和基本的流控制结构。不像 Java 通过描述语句来构建类的编译时间系统,JavaScript 支持运行时间系统,这种系统由少量表示数字值、逻辑值、字符串值的数据类型构成。JavaScript 有一种简单的、基于引例的对象模型,它能提供很多强有力的功能。JavaScript 也支持不需要作任何特定声明的函数,函数可作为对象的属性来对待,执行起来就像没有严格限制的类型化的方法一样。

Java 是一种面向对象的,为实现快速执行、类型安全的目的而设计的编程语言。类型安全意味着既不能把一个完整的 Java 应用程序剖析成可参考的对象,也不能通过转换其字节代码来查看它运行时所占据的内存数据。Java 的面向对象的模型指程序由专门的类和它们的方法构成。Java 的类的继承性和严格分类通常要求对象阶组的紧密结合。这些要求使得 Java 语言编程比 JavaScript 复杂得多。

与此相反,JavaScript 是与 HyperTalk、dBASE 类似的短小、精悍的描述语言。因为它们的语言更简单,有专门的内建功能,很少需要创建对象,所以这些描述语言的编程工具拥有更多的用户。

1.3.2 在 HTML 中嵌入 JavaScript

在 HTML 文档中我们可以通过以下几种方式嵌入 JavaScript:

- (1) 在 < SCRIPT > 标记中充当描述语句和函数;
- (2) 指定一个文件作为 JavaScript 源(而不是在 HTML 中嵌入 JavaScript 语句);
- (3) 指定一个 JavaScript 表达式作为 HTML 的一个属性值;
- (4) 在某些 HTML 标记中作为事件处理。

与 HTML 不同,JavaScript 区分大小写。

1. 使用 SCRIPT 标记

< SCRIPT > 标记是 HTML 中的一个扩展项,它能把 JavaScript 语句与 HTML 中的其他语句区分开。写法如下:

```
< SCRIPT >  
JavaScript 描述语句...  
< /SCRIPT >
```

在一个 HTML 文档中可多次出现 < SCRIPT > 标记。每个标记中都可放置许多 JavaScript 语句。

可选项 LANGUAGE 属性指定脚本语言和 JavaScript 的版本,如:

```
< SCRIPT LANGUAGE = "JavaScriptVersion" >  
    JavaScript 描述语句...  
< /SCRIPT >
```

JavaScriptVersion 指定 JavaScript 代码由下列哪个版本写成。

- < SCRIPT LANGUAGE = "JavaScript" > 代表 JavaScript for Navigator 2.0。
- < SCRIPT LANGUAGE = "JavaScript 1.1" > 代表 JavaScript for Navigator 3.0。

假如用户的浏览器不能支持在 LANGUAGE 属性中定义的 JavaScript 版本,在 < SCRIPT > 标记中的 JavaScript 描述语句部分将被浏览器忽略。例如:

· Navigator 2.0 执行在 < SCRIPT LANGUAGE = "JavaScript" > 标记中的代码;它忽略在 < SCRIPT LANGUAGE = "JavaScript1.1" > 标记中的代码。

· Navigator 3.0 既执行在 < SCRIPT LANGUAGE = "JavaScript" > 标记中的代码;也执行在 < SCRIPT LANGUAGE = "JavaScript1.1" > 标记中的代码。

假如 LANGUAGE 属性被省略,Navigator 2.0 假定 LANGUAGE = "JavaScript";Navigator 3.0 假定 LANGUAGE = "JavaScript1.1"。

用户可以用 LANGUAGE 属性写含有 Navigator 3.0 特性的代码,它在 Navigator 2.0 下照样可以无错误地运行,下面举几个使用 LANGUAGE 属性的例子。

例 1:分别为 JavaScript 1.0 和 JavaScript 1.1 在同一个文档中定义两次函数,以适用于不同的浏览器版本。代码如下:

```
< SCRIPT LANGUAGE = "JavaScript" >  
// 定义适合 1.0 的函数,如:doClick()  
< /SCRIPT >  
< SCRIPT LANGUAGE = "JavaScript1.1" >  
// 用 1.1 特性再定义上面的函数  
// 还可在此定义仅适用于 1.1 的函数  
< /SCRIPT >
```

```
< FORM ... >
```



```
< INPUT TYPE = "button" onClick = "doClick(this)" ... >
...
</FORM >
```

例 2: 使用两个不同版本的 JavaScript 的文档, 缺省情况下载入用 JavaScript 1.0 写成的文档, 当用户运行 Navigator 3.0 时, 用 **replace** 方法载入用 JavaScript 1.1 写成的文档。

```
< SCRIPT LANGUAGE = "JavaScript 1.1" >
// 如果为 Navigator 3.0, 用 mypage.html 代替下面的页
location.replace("js1.1/mypage.html")
</SCRIPT >
```

[从此往下是 JavaScript 1.0 写的页面...]

例 3: 这个例子演示如何通过测试 *navigator.userAgent* 属性的值来判断用户是否在使用 Navigator 3.0, 从而使代码有条件地执行 JavaScript 1.1 的特性。

```
< SCRIPT LANGUAGE = "JavaScript" >
if (navigator.userAgent.indexOf("3.0") != -1)
    jsVersion = "1.1"
else
    jsVersion = "1.0"
</SCRIPT >
```

[此后, 用 JavaScript 1.1 语句判断 *jsVersion == "1.1"* 是否成立]

例 4: 在一些情况下, 通过判断一些新属性(如: *navigator.javaEnabled* or *window.focus*) 的值是否为 null, 可得知 JavaScript 的版本。在 JavaScript 1.0 和 JavaScript 1.1 中, 未被定义的属性值为 null, 所以 JavaScript 1.1 的函数参考符号在 Navigator 3.0 下将是非 null 值。

```
if (navigator.javaEnabled != null && navigator.javaEnabled()) {
    //是 Navigator 3.0 版且能运行 Java, 可在这儿用 LiveConnect...
}
else {
    // 是 Navigator 2.0, 不能运行 Java
    Navigator 2.0 的语句
}
}
```

仅仅 Netscape Navigator 2.0 和以后的版本能解释 JavaScript 语句。为了确保其他浏览器忽略 JavaScript 代码, 把整个的 JavaScript 语句放在 HTML 的 comment 标记中, 在 comment 结束标记之前插入双斜杠(//)来指示仅限于 JavaScript 的内容:

```
< SCRIPT >
< ! -- Begin to hide script contents from old browsers.
JavaScript statements...
// End the hiding here. -- >
</SCRIPT >
```

因为通常的浏览器都将忽略不认识的标记, 所以不能解释 JavaScript 语句的浏览器将忽略开头和结尾的 SCRIPT 标记。而所有的脚本描述语句都被包含在 HTML 的 comment 标记中, 因此也将被忽略。Navigator 能正确地解释 SCRIPT 标记并忽略带有双斜杠(//)的语句。

下面再举一个用 JavaScript 语句输出文本的例子, 如:

```
< HTML >
```