



高等院校计算机类课程“十二五”规划教材

Java Web 程序设计

Java Web CHENGXU SHEJI

主 编◎郑 刚 涂立新

副主编◎孙国春 郭 辉 夏政伟 吴 燕



可下载教学资源

<http://www.hfutpress.com.cn>

<http://blog.csdn.net/crcr>

nwuchenrui@126.com



合肥工业大学出版社
HEBEI UNIVERSITY OF TECHNOLOGY PRESS

高等院校计算机类课程“十二五”规划教材

Java Web 程序设计

主 编 郑 刚 徐立新
副主编 孙国春 郭 辉 夏政伟 吴 燕
参 编 夏敏捷 邓 华 刘 兵 张西广

内容提要

本书介绍了 Java Web 开发中的主要相关技术及相关知识。全书分为 12 章,内容包括 Java Web 开发概述、搭建 Java Web 开发环境、Servlet 技术、JSP 技术、跟踪会话、过滤器 Filter、监听器 Listener、JSTL 标签库、自定义标签、数据库概述、JDBC 使用以及在 Java Web 开发中经常遇到的问题与解决的方法。

本书内容全面,结构清晰,语言通俗流畅,重难点突出,例题丰富,所有程序都能够直接运行。本书可作为大中专院校的计算机及相关专业的 Java Web 程序设计教材,也可作为计算机软件开发人员的参考书。

图书在版编目(CIP)数据

Java Web 程序设计/郑刚,徐立新主编. —合肥:合肥工业大学出版社,2013. 8
ISBN 978 - 7 - 5650 - 1438 - 3

I. ①J… II. ①郑…②徐… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 182543 号

Java Web 程序设计程序设计

郑 刚 徐立新 主编		责任编辑 汤礼广 石金桃	
出 版	合肥工业大学出版社	版 次	2013 年 8 月第 1 版
地 址	合肥市屯溪路 193 号	印 次	2013 年 8 月第 1 次印刷
邮 编	230009	开 本	787 毫米×1092 毫米 1/16
电 话	理工编辑部:0551-62903087	印 张	21.5
	市场营销部:0551-62903163	字 数	498 千字
网 址	www.hfutpress.com.cn	印 刷	合肥星光印务有限责任公司
E-mail	hfutpress@163.com	发 行	全国新华书店

ISBN 978 - 7 - 5650 - 1438 - 3

定价: 43.00 元

如果有影响阅读的印装质量问题,请与出版社市场营销部联系调换。

前言

近几年,Web 程序发展异常迅猛,Web 程序的开发也变得越来越复杂。实现 Web 程序的技术很多,但 Java 相关技术作为最流行的开发技术之一,在 Web 开发领域有着非常广泛的应用。利用 Java 开发 Web 应用程序所涉及的技术比较繁杂,这对需要掌握 Java Web 开发的人来说,确实有一定的困难。为帮助读者学好 Java Web 开发技术,能够熟练运用 Java 语言开发 Web 程序,在本书中我们对 Java Web 开发的基本技术与技能技术进行了有机结合,并采用最简单易懂的实例进行讲解。读者在学完本书内容后,一般来说,都可以掌握 Java Web 开发所涉及的基础知识和基本技术,并会搞清楚它们之间的联系,基本掌握或提升自己的 Java Web 开发水平。

本书主要介绍 Java Web 开发的基本知识,重点介绍 Java Web 开发的原理及概念,如 Servlet、JSP 与 JDBC,通过示例代码说明其功能和应用。

本书的特点

1. 论述严谨,体系完整

本书对基本理论进行了全面、准确的论述,便于读者形成完整的知识体系。本书在组织上紧扣 Java Web 开发的基础知识,从 Java Web 开发的基本概念、原理出发,介绍其实际应用,并进一步引入程序设计基础、设计模式与原则,因此显得结构合理、脉络清楚。

2. 入门快速,易教易学

本书突出“上手快、易教学”的特点。各章在详细阐述基本概念和要点的同时,还通过相应的范例来进一步地说明问题,以加深读者的理解。各章后面均附有一定数量的习题,便于读者复习。

3. 学以致用,注重能力

本书将实际开发经验融入基本理论之中,力求使读者在掌握基本理论的同时,获得实际开发的基本思想方法,并得到一定程度的项目开发实训,以培养学生独立开发较为复杂的系统的能力。

4. 深入浅出,螺旋上升

本书内容和示例的安排难点分散、前后连贯,并采用循序渐进的编写风格,层次清晰,步骤详细,便于学生理解和实现。

本书的内容

第1章:主要介绍了 Java Web 开发的一些基本概念以及相关技术的背景知识。

第2章:主要介绍了搭建 Java Web 开发环境、开发部署第一个 Java Web 程序。

第3章:主要介绍了 Servlet 规范的基本技术和高级特性。

第4章:主要介绍了 JSP 技术。

第5章:主要介绍了跟踪会话 Session 和 Cookie。

第6章:主要介绍了过滤器 Filter。

第7章:主要介绍了监听器 Listener。

第8章:主要介绍了 JSP 标签及 JSTL 标签库的使用。

第9章:主要介绍了自定义标签库的实现。

第10章:主要介绍了 JSP 开发中常见的问题与解决的方法。

第11章:主要介绍了数据库的基础知识、数据定义语言与结构化查询语句。

第12章:主要介绍了 Java 访问数据库的技术 JDBC 的用法。

本书由郑刚(河南艺术职业学院)、徐立新(空军航空大学)主编,孙国春(空军航空大学)、郭辉(空军航空大学)、夏政伟(许昌学院)、吴燕(江西农业大学计算机信息与工程学院)担任副主编,夏敏捷(中原工学院)、邓华(空军航空大学)、刘兵(空军航空大学)、张西广(中原工学院)参编。本书由陈锐负责统稿。

由于作者水平有限,加之时间仓促,书中难免存在一些错误和不足之处,恳请读者批评指正。

在使用本书过程中,如有好的建议和想法,或需本书的例题代码,可通过 nwuchenrui@126.com 与作者联系。

编 者

目 录

第 1 章 Java Web 开发概述	(1)
1.1 Web 背景知识	(1)
1.2 Web 开发技术简介	(4)
1.3 Web 应用程序开发	(9)
小 结	(14)
思考题	(14)
第 2 章 搭建 Java Web 开发环境	(15)
2.1 Java Web 开发环境概述	(15)
2.2 下载安装 Java 环境	(16)
2.3 下载安装 Tomcat	(21)
2.4 安装使用 Eclipse	(25)
2.5 第一个 Java Web 程序	(27)
2.6 部署到 Tomcat	(31)
小 结	(37)
思考题	(38)
第 3 章 深入 Servlet 技术	(39)
3.1 Servlet 概述	(39)
3.2 编写 Servlet	(42)
3.3 请求与响应	(47)
3.4 读取 web.xml 参数	(54)
3.5 提交表单信息	(63)
3.6 Servlet 生命周期	(77)
3.7 Servlet 之间的跳转	(83)
小 结	(89)
思考题	(89)

第 4 章 深入 JSP 技术	(90)
4.1 JSP 简介	(90)
4.2 JSP 语法	(94)
4.3 JSP 的流程控制	(98)
4.4 JSP 指令	(104)
4.5 JSP 行为	(109)
4.6 JSP 隐藏对象	(120)
4.7 EL 表达式	(126)
4.8 JSP 2 标签文件(Tag Files)	(129)
4.9 JSP 配置	(136)
小 结	(138)
思考题	(138)
第 5 章 跟踪会话 Session	(139)
5.1 Cookie 机制	(139)
5.2 Session 机制	(161)
5.3 Session 与 Cookie 的比较	(167)
小 结	(169)
思考题	(169)
第 6 章 过滤器 Filter	(170)
6.1 Filter 概述	(170)
6.2 Filter 经典案例	(173)
小 结	(181)
思考题	(181)
第 7 章 监听器 Listener	(182)
7.1 Listener 概述	(182)
7.2 Listener 分类	(183)
7.3 Listener 使用案例	(189)
小 结	(200)
思考题	(200)

第 8 章 标准标签库 JSTL	(201)
8.1 JSTL 概述	(201)
8.2 JSTL 的 core 标签库	(204)
8.3 JSTL 的 fmt 标签库	(213)
8.4 JSTL 的 SQL 标签库	(223)
8.5 JSTL 的 XML 标签库	(223)
8.6 JSTL 的 fn 方法库	(223)
小 结	(226)
思考题	(226)
第 9 章 自定义标签	(227)
9.1 编写自定义标签	(227)
9.2 嵌套的自定义标签	(239)
9.3 JSP 2. x 自定义标签	(245)
9.4 编写自定义方法	(249)
小 结	(252)
思考题	(252)
第 10 章 JSP 常用技巧	(253)
10.1 资源国际化	(253)
10.2 中文编码	(261)
10.3 错误处理机制	(264)
10.4 限制表单重复提交	(265)
小 结	(268)
思考题	(268)
第 11 章 数据库概述	(269)
11.1 数据库简介	(269)
11.2 MySQL 数据库	(270)
11.3 数据定义语言 DDL	(275)
11.4 数据操作语言 DML	(279)
11.5 MySQL 客户端工具	(282)
小 结	(286)

思考题	(286)
第 12 章 JDBC 详解	(287)
12.1 JDBC 简介	(287)
12.2 MySQL 中文编码	(293)
12.3 JDBC 基本操作	(295)
12.4 处理结果集	(312)
12.5 JDBC 高级应用	(330)
小 结	(335)
思考题	(335)
参考文献	(336)

第 1 章

Java Web 开发概述

随着 Internet 的发展,Web 应用程序在网络上更加广泛地进行传播和使用。绝大部分 Web 应用程序基于 B/S(浏览器/服务器)方式,用户只要通过任一款浏览器(如 IE 或者 Firefox)就可以使用 Web 应用程序。

1.1 Web 背景知识

在了解如何开发 Web 应用程序之前,有必要先了解一下这些应用程序的运行平台和环境。下面简单介绍涉及 Web 开发的相关知识,包括 Web 浏览器、Web 服务器、基本访问原理以及 HTTP 协议。

1.1.1 Web 浏览器

在众多 Web 浏览器中,比较普及和流行的浏览器有 Mozilla 基金会的 Firefox、Google 公司推出的 Chrome 和 Microsoft 公司的 IE(Internet Explorer)。其中,Firefox 和 Chrome 浏览器都能很好地支持最新的 HTML 标准及各种扩展功能,而且都能较好地支持 JavaScript 脚本语言;而 IE 浏览器早期一直处于垄断地位,长期以来没有太大的变化,功能中规中矩,安全漏洞不时出现,对新标准和特性支持不够完善,尽管新版本相对旧版本变化很大,但已经逐渐被 Chrome 和 Firefox 浏览器超越。

其他的浏览器还有 Opera、Safria、国产的傲游浏览器(Maxthon)、腾讯 TT 浏览器等。这些浏览器由于用户较少或基于 IE 内核二次开发,在 Web 开发中使用较少。

总之,浏览器是开发和使用 Web 的重要工具,它是一种浏览 Web 信息的计算机软件,通过访问 Web 服务器把信息显示在用户计算机上。

1.1.2 Web 服务器

服务器(Server)既可以是计算机硬件,有时又代表计算机服务器端软件。用户如果要区分它们,主要是从语境上去区分。

(1) 服务器是一种计算机硬件

当你向一个硬件销售商咨询购买服务器时,这个时候一般是指计算机硬件。服务器也是计算机,它的构成与普通个人计算机基本相同,也有 CPU、硬盘、内存、主板等,但服务器在某些方面(如多个 CPU、并行运算)做了优化和强化。

服务器应该算是一种高性能的计算机,它作为网络的节点,存储、处理网络上的数据和信息,因此也被称为网络的灵魂。

(2) 服务器是一种计算机软件

如果别人跟你提起 Apache、IIS 服务器、Java 服务器、.NET 服务器等这样的名词时,一般都是指一种安装在服务器上的计算机软件。当用户使用计算机上网时,虽然是访问服务器硬件,但其实是服务器硬件上安装了相应的服务器软件,这些服务器软件负责接收用户的访问请求,并根据请求经过计算将数据返回给用户的客户端(浏览器)。

常见的服务器软件分为两类:一类是 Web 服务器;另一类是应用程序服务器(App Server)。Web 服务器提供 Web 信息服务,主要支持的协议是 HTTP;而应用程序服务器可以通过很多协议(包括 HTTP)来为应用程序提供业务逻辑功能。目前,多数应用程序服务器都具备 Web 服务器的功能。Apache 和 IIS 服务器是常用的 Web 服务器软件;Java 服务器、.NET 服务器则是应用程序服务器软件。

1.1.3 Web 访问基本原理

回想一下我们平时浏览网页的时候,浏览器和服务器端都发生了什么变化,网站是如何实现请求和响应功能的。图 1-1 清晰地显示了浏览器访问 Web 服务器的整个过程。

(1) 用户打开浏览器(如 IE、Firefox 等),输入网站的 URL 地址,也就是通常所说的网址。这个地址告诉浏览器要访问互联网中的哪台主机。

(2) 浏览器寻找到指定的主机之后,向 Web 服务器发出请求(request)。

(3) Web 服务器接受请求并做出相应的处理,生成处理结果,大多数生成 HTML 格式,也有其他响应的格式。

(4) 服务器把响应的结果返回发送给浏览器。

(5) 浏览器接收到对应的响应结果后,在浏览器中显示响应结果,比如 Web 页面。

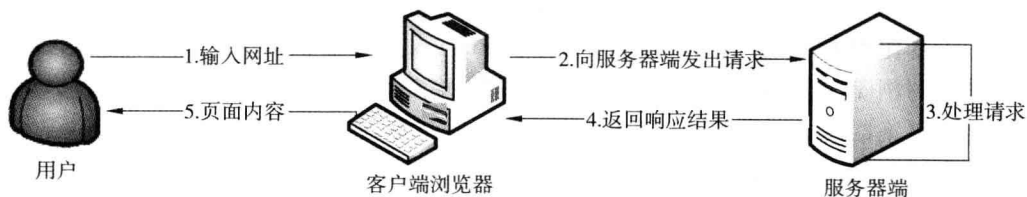


图 1-1 浏览器访问 Web 服务器的过程

1.1.4 HTTP 超文本传输协议

知道了浏览器与 Web 服务器之间的交互关系之后,再来认识一下负责浏览器与

Web 服务器之间交互的桥梁:HTTP 超文本传输协议。HTTP 全称为 Hyper Text Transfer Protocol,主要用来定义客户端和服务器的通信规范。

客户端通过 HTTP 网络协议从服务器上获取网页文档。为了让客户端能够定位到服务器中的资源,使用 URL 定义 Web 服务器中的每个网页文件。那么究竟什么是 URL? URL 全称为 Uniform Resource Locator,意思是统一资源定位符。URL 地址中包含网络协议、服务器主机名(IP 地址)、文件(其他资源)路径和端口号。

说明:在“http://127.0.0.1:8080/test/test.html”这个地址中,“http”为网络协议,“127.0.0.1”为服务器地址,“8080”为端口号,“/test/test.htm”为文件资源地址。

在复杂的企业应用中,可能会在同一个服务器上安装多个服务器软件,为了管理方便不发生冲突,这些服务器软件通过不同的端口来提供服务。Web 服务器就是通过监听某一端口,接收此端口的请求来提供服务的。一般情况下 Web 服务器主要对 TCP 80 端口(WWW 的默认端口)进行监听。

说明:80 为 Web 服务端口,访问时,URL 地址可省略默认的端口 80。如果不是 80,则必须在 URL 中添加端口号。很多服务都有规定的端口,如 FTP 为 21、SMTP 为 25、数据库 MySQL 的默认端口是 3306。

下面介绍基于 HTTP 协议的客户端/服务器模式的信息交换过程,它分为 4 个过程:建立连接、发送请求信息、发送响应信息、关闭连接。

- 建立连接:连接的建立是通过申请套接字(Socket)实现的。客户打开一个套接字并把它约束在一个端口上,如果成功,就相当于建立了一个虚拟文件。以后就可以在该虚拟文件上写数据并通过网络向外传送。

- 发送请求:打开一个连接后,客户机把请求消息送到服务器的相关端口上,完成提出请求动作。HTTP/1.0 请求消息的格式为:

请求消息 = 请求行(通用信息|请求头|实体头) CRLF [实体内容]

请求行 = 方法 请求 URL HTTP 版本号 CRLF

方法 = GET|HEAD|POST|扩展方法

URL = 协议名称 + 宿主名 + 目录与文件名

请求行中的方法描述指定资源中应该执行的动作,常用的方法有 GET、HEAD 和 POST。

- 发送响应:服务器在处理完客户的请求之后,要向客户机发送响应消息。HTTP/1.0 的响应消息格式如下:

响应消息 = 状态行(通用信息头|响应头|实体头) CRLF [实体内容]

状态行 = HTTP 版本号 状态码 原因叙述

状态码表示响应类型:

- 1——保留;
- 2——请求成功地接收;
- 3——完成请求客户需进一步细化请求;
- 4——客户错误;
- 5——服务器错误。

响应头的信息包括:服务程序名,通知客户请求的 URL 需要认证,请求的资源何时能使用。

- 关闭连接:客户和服务器双方都可以通过关闭套接字来结束 TCP/IP 对话。

1.2 Web 开发技术简介

Web 是一种典型的分布式应用架构。Web 应用中的每一次信息交换都要涉及客户端和服务端两个层面。因此,Web 开发技术大体上也可以被分为客户端技术和服务端技术两大类。下面介绍 Web 开发需要了解的客户端和服务端的一些相关技术。

1.2.1 HTML 概述

HTML(HyperText Markup Language)即超文本标记语言,它是一种用于编写超本文档的标记语言。自从 1990 年首次应用于网页编辑后,HTML 迅速崛起成为网页编辑的主流语言。几乎所有的网页都是由 HTML 或以其他程序语言嵌套在 HTML 中编写的。HTML 并不是一种程序语言而是一种结构语言,它具有平台无关性。无论用户使用何种操作系统(Windows、DOS、Unix 等),只要有相应的浏览器程序,就可以运行 HTML 文档。

HTML 的主要语法是元素和标签。元素是符合 DTD(文档类型定义)的文档组成部分,如 title(文档标题)、img(图像)、table(表格)等等。元素名不区分大小写。HTML 用标签来规定元素的属性和它在文档中的位置。标签分为单独出现的标签和成对出现的标签两种。大多数标签是成对出现的,由首标签和尾标签组成。首标签的格式为<元素名>,尾标签的格式为</元素名>。成对标签用于规定元素所包含的范围,如<title>和</title>标签用来界定标题元素的范围,也就是说,<title>和</title>之间包含的部分是该 HTML 文档的标题。单独标签的格式为<元素名>,它的作用是在相应的位置插入元素。如
标签表示在该标签所在位置插入一个换行符。下面来看一个简单的例子。

```
<html>
  <head>
    <title>HTML 范例</title>
  </head>
  <body>
    HTML is no real language such as C + + or Pascal,<br>
    it is just a system for describing documents. <br>
    A WWW browser interpret the<BR>
    HTML - code and display it.
  </body>
</html>
```

可见,一个 HTML 文档的基本结构如下:

<html> —— 标志着 HTML 文档开始

```
<head>
    <title>文档标题</title>
</head>
<body>
    文档的主体部分
</body>
</html> —— 标志着 HTML 文档结束
```

`<head>`与`</head>`之间包含的部分是 HTML 文档的文档头部分,用以说明文档的标题和整个文档的一些公共属性。`<body>`与`</body>`之间包含的部分是 HTML 文档的主体部分。

1.2.2 CSS 概述

CSS 是 Cascading Style Sheets(级联样式表)的缩写。CSS 是一种样式表语言,用于为 HTML 文档定义布局。例如,CSS 涉及字体、颜色、边距、高度、宽度、背景图像、高级定位等方面。CSS 为网站添加布局效果提供了更多选择,而且更为精确、完善。现在所有浏览器都支持 CSS。

随着网页逐渐流行起来,网页设计者们开始寻求为网页增添布局的可能性。为此,当时的浏览器厂商们(网景公司和微软公司)发明了一些新的 HTML 标签(如`<font>`等)以引入新的布局——而非新的结构。这也导致了原本用于进行文本结构化的标签(如`<table>`等)越来越多地被误用于进行页面的布局。许多新的布局标签(如`<blink>`)只有一种浏览器支持。

CSS 的发明正是为了改善这一状况,它为 Web 设计师们提供了完善的、所有浏览器都支持的布局能力。文档的表现样式与内容的分离,也令网站维护容易了许多。HTML 用于结构化内容;CSS 用于格式化结构化的内容。

CSS 是 Web 设计界的一次革命。CSS 的具体优点包括:通过单个样式表控制多个文档的布局;更精确的布局控制;为不同的媒体类型(屏幕、打印等)采取不同的布局。

CSS 应用一般有 3 种类型:外部样式表、内部样式表和行内样式表。

行内样式表指的是在 HTML 标签中通过 `style` 属性对 HTML 元素进行样式控制。

```
<html>
  <head>
    <meta http-equiv = "Content-Type" content = "text/html;charset = UTF-8"/>
    <title>CSS 行内样式示例</title>
  </head>
  <body style = "background-color: # CFF;font-size:18px;color: # F39">
    这是一个 CSS 行内样式表
  </body>
</html>
```

内部格式表是指在 HTML 文档内部定义的 CSS 样式表,内部样式表主要由 3 部分组成:标签名称、属性和属性值。其使用格式为:标签名称{属性 1:值 1;属性 2:值 2...}。

```
<html>
  <head>
    <meta http-equiv = "Content-Type" content = "text/html;charset = UTF-8"/>
    <title>CSS 行内样式示例</title>
    <style type = "text/css">
      <!--
        body. td. th {
          font-family: 隶书, 华文中宋;
          font-size: 20px;
          color: # F36;
        }
        body {
          background-color: # FFC;
          margin-top: 50px;
          margin-left: 50px;
        }
        table {
          width: 70 % ;border: 0;align: center;cellspacing: 1
        }
      -->
    </style>
  </head>
  <body>
    其他代码略。
  </body>
</html>
```

外部样式表其实就是将内部样式表放入一个扩展名为 .css 的文件中,将定义样式的代码彻底地从 HTML 文档中分离出来。在 HTML 文档的<head>标签加入<link href = “ * .css” ref = “stylesheet” type = “text/css”>的方法导入外部 css 样式文件。

```
<html>
  <head>
    <meta http-equiv = "Content-Type" content = "text/html;charset = UTF-8"/>
    <title>CSS 外部样式示例</title>
    <link href = "css/default.css" ref = "stylesheet" type = "text/css">
  </head>
  <body>
    其他代码略。
  </body>
</html>
```

1.2.3 JavaScript 概述

JavaScript 早期是由 Netscape(网景)公司研发出来的一种在 Netscape 浏览器上执行的程序语言。从结构来说,JavaScript 是一种基于对象和事件驱动的安全脚本语言,它还包括一般语言具备的操作数、控制流程等结构组件。含有 JavaScript 的网页不能单独执行,必须由浏览器或服务器解释执行。

开发 JavaScript 的最初动机是想减轻服务器数据处理的负荷,在网页上能够完成诸如显示时间、动态广告、处理表单传送数据等工作,设计出交互式的网页内容,从而可以开发客户端的应用程序。随着 JavaScript 所支持的功能日益增多,许多网页编制人员利用它来进行动态网页的设计,这使得 JavaScript 成为浏览器的通用语言。

在上面已经提到 JavaScript 只能在 HTML 脚本中才能得以执行。那么,如何在 HTML 语言中植入一个 JavaScript 脚本呢? 以下是一个 JavaScript 脚本结构示例:

```
<html>
  <head>
    <meta http-equiv = "Content-Type" content = "text/html;charset = UTF-8"/>
    <title>CSS 外部样式示例</title>
    <script language = "JavaScript">
      ...
    </script>
  </head>
  <body>
    ...
    <script language = "JavaScript">
      alert("hello, javascript.");
    </script>
    ...
  </body>
</html>
```

language 指定了一个 JavaScript 语言。但也可以不用写该属性,因为目前大部分浏览器都将其设为默认值。一段 JavaScript 代码可以放置于 HTML 中的任意部位,但多数情况下将其放于<head>...</head>区段中,因为一些代码可能需要在页面装载起始就开始运行。但放置在任何位置都是可以的,甚至可以在 HTML 外部装入一个 JavaScript 程序,如下面的代码所示:

```
<script language = "JavaScript" src = "url"></script>
```

这里的“url”就是一个外部的 JavaScript 程序,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。

JavaScript 代码结束后应在末尾添加一个分号,表示一个完整的句子,也可以将其省略,这样并不影响程序的运行。程序中的//和<!-- -->则是注释的标记。

下面是 JavaScript 的一个简单实例。代码如下：

```
<html>
<head>
  <title>这是一个 JavaScript 程序的例子! </title>
</head>
<body>
  <script language = "JavaScript">
    document.write("<h1>欢迎学习 JavaScript! </h1>");
  </script>
  <img src = "E:\javascript\001.gif">
</body>
</html>
```

常常会有人问到 JavaScript 与 Java 有什么区别,甚至有人误认为 JavaScript 就是 Java。虽然 JavaScript 与 Java 确实有一定的联系,但也有很大的区别。JavaScript 与 Java 的区别主要表现在以下几个方面:

- JavaScript 与 Java 是由不同的公司开发的不同产品。JavaScript 是 Netscape 公司的产品,其目的是为了扩展 Netscape Navigator 功能。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++,它需要多种编译器和支持文件才能运行。

- JavaScript 是基于对象的,是一种脚本语言。它基于对象和事件驱动,因而它本身提供了非常丰富的内部对象供设计人员使用。而 Java 是面向对象的,即 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 与 Java 在浏览器中所执行的方式不一样。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,即 JavaScript 语句本身随 Web 页面一起下载,由浏览器解释执行。而 Java 的源代码在传递到客户端执行之前,必须经过编译,所以客户端上必须具有相应平台上的仿真器或解释器,它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。

1.2.4 XML 概述

XML(eXtensible Mark Language,可扩展标记语言)在 1998 年 2 月成为 W3C 的推荐标准。XML 不是编程语言,是一种用来描述语义和数据结构的标记语言。XML 主要用于存储数据、交换数据、共享数据,还可用于创建新的标记语言。XML 具备了作为通用数据描述语言的优秀特征,可以使计算机数据在不同的计算机平台和程序之间方便、快捷、无障碍地转移和流动,是当今处理分布式结构信息的有效工具。

XML 是元标记语言,具有自描述性、独立于平台和良好的移植性。XML 作为一种简单的数据存储语言,使用一系列简单的标记描述数据,而这些标记可以用方便的方式建立,保存在任何可以存储文本的文档中。虽然 XML 占用的空间比二进制数据要更多,但 XML 极其简单,易于掌握和使用。