

高等学校计算机专业规划教材

Java Web 程序设计与开发

马月坤 赵全明 编著

清华大学出版社



高等学校计算机专业规划教材

Java Web 程序设计与开发

马月坤 赵全明 编著

清华大学出版社
北京

内 容 简 介

本书介绍用 Java 语言进行 Web 程序设计与开发的技术与方法。全书共 16 章,包括 Java Web 入门、HTML 和 JavaScript 页面技术、JSP 编程、JavaBean 和 Servlet 开发、数据库访问、EL 和 JSTL 技术、Ajax、MVC、Struts 及基于 Struts 的 Java Web 应用开发。本书内容由浅入深,书中所有知识都结合具体示例进行介绍,涉及的程序代码都给出了详细的注释,可以使读者轻松领会使用 J2EE 体系中的相关技术进行 Web 程序开发的精髓,提高开发技能。在每一章的最后提供了一些案例,供读者训练。

本书适合作为高等院校计算机相关专业教材,也可供 Web 应用开发者学习。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java Web 程序设计与开发/马月坤,赵全明编著. —北京:清华大学出版社,2016

高等学校计算机专业规划教材

ISBN 978-7-302-44647-7

I. ①J… II. ①马… ②赵… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 179441 号

责任编辑:龙启铭 战晓雷

封面设计:何凤霞

责任校对:梁毅

责任印制:宋林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:22

字 数:522 千字

版 次:2016 年 10 月第 1 版

印 次:2016 年 10 月第 1 次印刷

印 数:1~2000

定 价:39.00 元



Java Web 开发技术以其开放性、灵活性、安全性和成熟度赢得了很大的市场,成为 Web 项目开发的重要技术手段之一。

本书主要介绍用 Java 语言进行 Web 程序设计与开发的相关技术和方法,全书共 16 章,内容包括 Java Web 应用概述、Java Web 开发环境配置、HTML 编程基础、JavaScript 编程技术、JSP 编程基础、JSP 内置对象、JSP 标签、JSP 访问数据库、JSP 和 JavaBean、JSP 自定义标签、Servlet 程序设计、Servlet 过滤器设计、EL 和 JSTL、Ajax 技术、MVC 和 Struts、Struts 2 基础开发。

本书采用以任务为导向的项目式体例编写,内容由浅入深,涵盖全面。书中所有知识都结合具体示例进行介绍,易于学生理解和接受。书中的程序代码都给出了详细的注释,可以使读者轻松领会使用 J2EE 体系中的相关技术进行 Web 程序开发的精髓,提高开发技能。每一章的最后提供了习题,以供读者复习使用。

本书由华北理工大学马月坤、赵全明编著。研究生黄东欢参与了本书的编写工作。

由于作者水平有限,书中错误在所难免,欢迎广大读者批评指正。

作者

2016 年 5 月

第 3 章 HTML 编程基础 / 24

3.1 HTML 文件执行过程	24
3.2 HTML 文件结构	27
3.2.1 HTML 文件基本构成	27
3.2.2 HTML 中的元素与标记	28
3.3 HTML 常用标记	28
3.3.1 HTML 图像标记	28
3.3.2 HTML 超链接	31



目录

第 1 章	Java Web 应用概述	/1
1.1	主流 Web 技术介绍	1
1.1.1	JSP 编程技术	1
1.1.2	ASP 编程技术	2
1.1.3	.NET 编程技术	2
1.1.4	PHP 编程技术	5
1.2	Java Web 技术介绍	6
1.2.1	Java Web 相关技术构成	6
1.2.2	Java Web 技术特点	10
	本章小结	11
	课后习题	11
第 2 章	Java Web 开发环境配置	/12
2.1	Tomcat 安装与配置	12
2.1.1	Tomcat 简介	12
2.1.2	Tomcat 的安装与配置	13
2.2	Eclipse 安装配置	17
2.2.1	Eclipse 简介	17
2.2.2	Eclipse 安装环境搭建	18
	本章小结	23
	课后习题	23
第 3 章	HTML 编程基础	/24
3.1	HTML 文件执行过程	24
3.2	HTML 文件结构	27
3.2.1	HTML 文件基本构成	27
3.2.2	HTML 中的元素与标记	28
3.3	HTML 常用标记	28
3.3.1	HTML 图像标记	28
3.3.2	HTML 超链接	31

3.3.3	HTML 换行、分段与字体标记	33
3.3.4	HTML 列表标记	38
3.3.5	span 和 div 标记	44
3.4	HTML 中的表格	47
3.4.1	表格基本结构	47
3.4.2	表格常用属性	53
3.4.3	其他表格属性	55
3.5	HTML 表单	56
3.5.1	表单的一般格式	56
3.5.2	<input> 标记	58
3.5.3	<button> 标记	67
3.5.4	<select> 标记	68
3.5.5	<option> 标记	70
3.5.6	<textarea> 标记	71
3.5.7	表单应用示例	72
3.6	HTML 的窗口切分	73
3.6.1	框架网页	73
3.6.2	浮动的网页窗口	76
3.7	本章案例	78
	本章小结	79
	课后习题	79
第 4 章 JavaScript 编程技术 /80		
4.1	JavaScript 简介	80
4.2	JavaScript 语法	81
4.2.1	JavaScript 代码的加入	81
4.2.2	数据类型	82
4.3	JavaScript 事件处理	84
4.3.1	异步事件与线程	85
4.3.2	常见事件特点	86
4.4	网页 DOM 模型	87
4.4.1	DOM 树简介	88
4.4.2	DOM 标记对象操作	89
4.5	JavaScript 内置对象	92
4.5.1	内置对象的层次及主要功能	92
4.5.2	window 对象	93
4.5.3	history 和 location 对象	102
4.5.4	screen 对象	104



4.6	本章案例	106
	本章小结	107
	课后习题	107
第 5 章 JSP 编程基础 /109		
5.1	第一个 JSP 页面	109
5.2	JSP 基本语法	111
5.2.1	JSP 声明语法	111
5.2.2	JSP 代码脚本	113
5.2.3	JSP 脚本注释	114
5.2.4	JSP 表达式	116
5.3	本章案例	117
	本章小结	118
	课后习题	119
第 6 章 JSP 内置对象 /120		
6.1	内置对象概述	120
6.2	内置对象的作用范围	121
6.2.1	page 范围	122
6.2.2	request 范围	123
6.2.3	session 范围	126
6.2.4	application 范围	127
6.3	内置对象	128
6.3.1	request 内置对象	128
6.3.2	response 内置对象	136
6.3.3	out 内置对象	140
6.3.4	page 内置对象	142
6.3.5	pageContext 内置对象	143
6.3.6	session 内置对象	145
6.3.7	application 内置对象	149
6.4	本章案例	152
	本章小结	155
	课后习题	155
第 7 章 JSP 标签 /157		
7.1	JSP 指令元素	157
7.1.1	page 指令	157
7.1.2	include 指令	164



7.2	JSP 动作标签	166
7.2.1	动作标签概述	166
7.2.2	include 动作标签	166
7.2.3	forward 动作标签	168
7.3	本章案例	168
	本章小结	170
	课后习题	171
第 8 章	JSP 访问数据库 /172	
8.1	JDBC 简介	172
8.2	建立 ODBC 数据源	173
8.3	JDBC 操作	176
8.3.1	添加数据操作	176
8.3.2	删除数据操作	180
8.3.3	修改数据操作	183
8.3.4	查询数据操作	187
8.4	PreparedStatement 对象	191
8.5	事务	192
8.6	使用厂商驱动程序进行数据库连接	194
8.7	本章案例	195
	本章小结	198
	课后习题	199
第 9 章	JSP 和 JavaBean /200	
9.1	在 JSP 中调用类	200
9.1.1	JSP 调用内部类	200
9.1.2	JSP 调用外部类	201
9.2	JavaBean 组件	202
9.2.1	JavaBean 的创建	203
9.2.2	JSP 页面中使用 JavaBean 对象	204
9.2.3	JavaBean 属性范围	208
9.2.4	表单和 JavaBean 组件交互	210
9.3	本章案例	213
	本章小结	215
	课后习题	215
第 10 章	JSP 自定义标签 /217	
10.1	JSP 自定义标签概述	217



10.1.1	自定义标签	217
10.1.2	相关概念	217
10.1.3	自定义标签的优势	218
10.1.4	JSP 自定义标签格式	219
10.1.5	自定义标签处理过程	220
10.2	标签库自定义基础	220
10.2.1	开发自定义标签涉及的接口与类	220
10.2.2	标签库的编写	221
10.3	简单的自定义开发案例	223
10.3.1	创建标签处理类	223
10.3.2	创建标签库描述文件	224
10.3.3	配置 web.xml 及在 JSP 页面引用自定义标签	225
10.4	几种标签的开发	225
10.4.1	空标签	225
10.4.2	带属性的标签	226
10.4.3	带标签体的标签	228
10.4.4	使用动态属性	229
10.5	本章案例	230
	本章小结	232
	课后习题	232
第 11 章 Servlet 程序设计 /233		
11.1	Servlet 技术简介	233
11.1.1	Servlet 的概念	233
11.1.2	第一个 Servlet 程序	235
11.2	Servlet 工作机制	237
11.2.1	Servlet 请求处理	237
11.2.2	Servlet 的生命周期	239
11.3	Servlet 程序设计基础	241
11.3.1	Servlet API	241
11.3.2	有关 Servlet 执行方式的配置	242
11.3.3	Servlet 与表单交互的方法	242
11.3.4	用 Servlet 控制会话	244
11.4	Servlet 程序设计过程	245
11.5	Servlet 和 JSP 的比较	246
11.6	本章案例	247
	本章小结	249
	课后习题	250



第 12 章 Servlet 过滤器设计 /251

- 12.1 过滤器的基本结构..... 251
 - 12.1.1 什么是过滤器..... 251
 - 12.1.2 过滤器工作原理..... 252
- 12.2 Servlet 过滤器 API 252
 - 12.2.1 Filter 接口 253
 - 12.2.2 FilterChain 接口 253
 - 12.2.3 FilterConfig 接口 253
 - 12.2.4 过滤器配置..... 254
- 12.3 Servlet 过滤器 254
 - 12.3.1 建立 Servlet 过滤器的基本步骤 254
 - 12.3.2 Servlet 过滤器编程案例 255
- 本章小结..... 257
- 课后习题..... 257

第 13 章 EL 和 JSTL /258

- 13.1 表达式语言..... 258
 - 13.1.1 为什么需要表达式语言..... 258
 - 13.1.2 表达式语言基本语法..... 259
- 13.2 基本运算符..... 259
 - 13.2.1 算术运算符..... 259
 - 13.2.2 关系运算符..... 260
 - 13.2.3 逻辑运算符..... 260
 - 13.2.4 “.”和“[]”运算符..... 260
 - 13.2.5 其他运算符..... 261
- 13.3 数据访问..... 262
 - 13.3.1 对象的作用域..... 262
 - 13.3.2 访问 JavaBean 263
 - 13.3.3 访问集合..... 264
 - 13.3.4 其他隐含对象..... 265
- 13.4 JSTL 技术 266
- 13.5 核心标签库..... 266
 - 13.5.1 核心标签库介绍..... 266
 - 13.5.2 用核心标签进行基本数据操作..... 267
 - 13.5.3 用核心标签进行流程控制..... 271
- 13.6 XML 标签库简介 273
- 13.7 国际化标签库简介..... 275



13.8	数据库标签库简介	278
13.9	函数标签库简介	279
	本章小结	281
	课后习题	281

第 14 章 Ajax 技术 /283

14.1	Ajax 概述	283
14.1.1	为什么需要 Ajax 技术	283
14.1.2	AJax 技术介绍	285
14.2	Ajax 开发	288
14.2.1	Ajax 核心代码	289
14.2.2	API 解释	290
14.3	本章案例	292
	本章小结	297
	课后习题	297

第 15 章 MVC 和 Struts /299

15.1	MVC 设计模式	299
15.1.1	MVC 简介	299
15.1.2	JSP Model1 和 JSP Model2	301
15.2	Struts 框架的基本原理	302
15.2.1	Struts 框架简介	302
15.2.2	Struts 框架原理	302
15.3	Struts 框架主要成分	304
15.3.1	ActionForm	304
15.3.2	Struts 标签库	304
15.4	基于 Struts 框架的应用开发	313
15.4.1	导入 Struts 框架	313
15.4.2	编写 JSP 页面	316
15.4.3	编写 ActionForm	317
15.4.4	编写 Action	318
15.4.5	创建 JavaBean 模型组件	319
15.4.6	配置 Struts-config.xml 文件	319
15.4.7	测试	320
	本章小结	321
	课后习题	321

(3) 强大的可伸缩性,从只有一个小的小的 jar 文件就可以运行 Servlet/JSP 到由多台服务器进行集群和负载均衡,到多台应用进行事务处理,消息处理,适用于各种规模的应用。



第 16 章 Struts 2 基础开发 /323

16.1 Struts 2 简介 323

16.2 Struts 2 基本原理 324

 16.2.1 环境配置 324

 16.2.2 Struts 2 原理 326

16.3 Struts 2 框架 327

 16.3.1 Action 接口 327

 16.3.2 Struts 2 的标签 329

 16.3.3 Struts 2 拦截器 332

16.4 Struts 2 框架开发实例 334

 16.4.1 编写 JSP 页面 334

 16.4.2 编写 Action 335

 16.4.3 配置 Action 336

 16.4.4 测试 337

 本章小结 337

 课后习题 337

参考文献 /339

第 1 章

Java Web 应用概述

本章要点

- 掌握主流 Web 应用开发手段的种类,并比较它们的优缺点。
- 了解 Java Web 技术基本概念。

学习目标

了解 Java Web 及相关技术概况,比较各种技术的不同,加深对 Java Web 技术特点的理解。

Java Web 应用程序指以 JSP、Servlet、JavaBean 等 Java 技术进行开发并运行在 Web 服务器上的应用程序。这类应用程序使用浏览器发起访问,服务器接收到浏览器的访问请求后,根据请求资源的不同返回不同的结果。这类程序也被称为基于浏览器-服务器结构的程序,简称 B/S 程序。本章将比较几种主流 Web 程序开发技术,并重点介绍 Java Web 技术的优势。

1.1 主流 Web 技术介绍

随着网络的快速发展,基于 B/S 结构的应用程序已成为当前最为流行的应用程序模式。目前支持 B/S 程序开发的技术有很多。比如 ASP 技术、.NET 技术、PHP 技术、JSP 技术等,下面对它们的特点进行比较。

1.1.1 JSP 编程技术

JSP(Java Server Page)技术基于 Java,它是一个特别的 Java 语言,加入了一个特殊的引擎,这个引擎将 HTTPServlet 这个类的一些对象自动进行初始化以便让用户使用。同时这个引擎又引入了 Java Servlet Code(.java 文件)和 Java Runtime Bin Code(.class 文件)。

JSP 有以下优势:

- (1) 一次编写,到处运行。
- (2) 系统的多平台支持。系统可以在任意环境下开发,在任意环境下进行系统部署,在任意环境下扩展。
- (3) 强大的可伸缩性。从只有一个小的 jar 文件就可以运行 Servlet/JSP 到由多台服务器进行集群和负载均衡,到多台应用进行事务处理、消息处理,适用于各种规模的应用

开发。

(4) 多样化和功能强大的开发工具支持。这一点与 ASP 很像,Java 已经有了许多非常优秀的开发工具,而且很多可以免费得到,并且其中不少已经可以顺利地运行于多种平台之上。

1.1.2 ASP 编程技术

ASP(Active Server Pages)是微软公司推出的一种用以取代 CGI(Common Gateway Interface,通用网关接口)的技术,它在 IIS 2.0 上首次推出,在 IIS 3.0 上发扬光大,成为服务器端应用程序的热门开发工具。Internet 上的许多基于 Windows 平台的 Web 网站使用 ASP 来替换 CGI。

简单地讲,ASP 是一个位于服务器端的脚本运行环境,通过这种环境,用户可以创建和运行动态的交互式 Web 服务器应用程序,如交互式动态网页,包括使用 HTML 表单收集和處理信息、上传与下载等。通常情况下,用户通过浏览器看到的网页大多是静态的,目前 Internet 上的许多网站仍然提供静态(static)的主页内容。所谓“静态”,指的就是网站的主页内容是“固定不变”的。当浏览器通过 Internet 的 HTTP(Hypertext Transport Protocol)协议向网站服务器(Web Server)请求提供主页的内容时,网站服务器收到请求后,就传送原已设计好的静态 HTML 文件数据给浏览器。一个“静态”的网站,若要更新主页内容,必须手动更新其 HTML 的文件数据。而随着 Web 应用的发展,用户希望能够看到根据要求而动态生成的主页,例如响应用户查询数据库的要求、生成报表等。

ASP 是一种脚本式的解释型程序,其主要支持的脚本语言有 JavaScript 和 VBScript。这类程序的运行需要使用 IIS(Internet Information Server)的服务器作为支持。由于 IIS 受到平台限制,所以 ASP 技术主要应用于 Windows 平台上,这类程序的跨平台部署十分困难。ASP 程序由 HTML 代码和 ASP 脚本代码构成,共同完成显示样式的控制以及程序逻辑的实现,这使得 ASP 程序在开发和维护上较后面介绍的几类 Web 开发技术更为困难。也正是由于这些原因,ASP 技术不太适合逻辑结构复杂的 B/S 程序开发。

ASP 是解释型程序,每次接收到用户请求时,服务器都要对 ASP 代码重新解释、运行,从而导致 ASP 程序执行效率不高。当网站访问量过大时,这种技术缺陷带来的弊端就会十分明显,而且由于其自身特点,ASP 程序结构可被优化的程度不高。

另外,ASP 技术属于一种半过程、半面向对象的语言环境,不支持面向对象中的许多重要概念和方法,这与当前主流的开发模式格格不入,势必被其他技术取代。

1.1.3 .NET 编程技术

.NET 是 Microsoft XML Web Service 平台。XML Web Service 允许应用程序通过 Internet 进行通信和共享数据,而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建 XML Web Service 并将这些服务集成在一起的技术。对个人用户的好处是无缝的、吸引人的体验。

.NET 是微软公司用来实现 XML、Web Service、SOA(service-oriented architecture,

面向服务的体系结构)和敏捷性的技术。对技术人员,想真正了解什么是.NET,必须先了解.NET 技术出现的原因和它想解决的问题,必须先明确为什么需要 XML、Web Service 和 SOA。技术人员一般将微软公司看成一个平台厂商。微软公司搭建技术平台,而技术人员在这个技术平台之上创建应用系统。从这个角度来看,.NET 也可以如下来定义:.NET 是微软公司的新一代技术平台,为敏捷商务构建互连互通的应用系统,这些系统是基于标准的、连通的、适应变化的、稳定的和高性能的。从技术的角度来看,.NET 应用是一个运行于.NET Framework 之上的应用程序(更精确地说,.NET 应用是一个使用.NET Framework 类库来编写,并运行于公共语言运行时(Common Language Runtime)之上的应用程序)。如果一个应用程序与.NET Framework 无关,它就不能称为.NET 程序。比如,仅仅使用了 XML 并不是.NET 应用,仅仅使用 SOAP SDK 调用一个 Web Service 也不是.NET 应用。.NET 是基于 Windows 操作系统运行的、应用于互联网的分布式操作平台。

Microsoft .NET 包括的技术领域如图 1-1 所示。

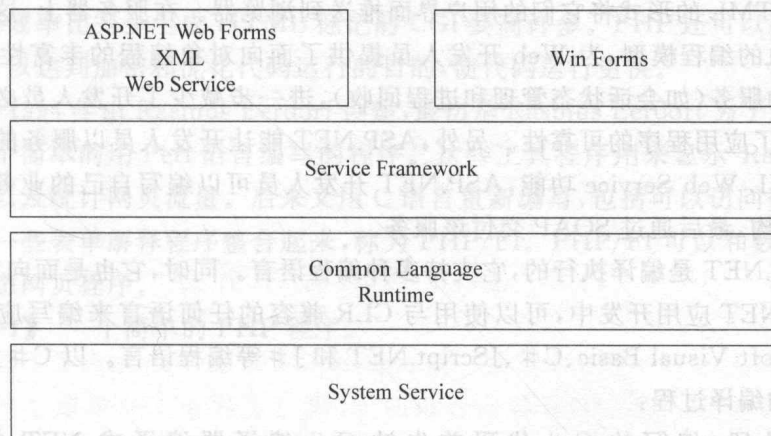


图 1-1 Microsoft .NET 技术领域

其中,.NET 框架是一个多语言组件开发和执行环境,它提供了一个跨语言的统一编程环境。.NET 框架的目的是便于开发人员更容易地建立 Web 应用和 Web 服务,使得 Internet 中的各种应用之间可以使用 Web 服务进行沟通。开发人员可以将远程应用的服务和单机应用结合起来,组成一个全新的应用。

.NET 语言和语言工具支持多种编程语言,如表 1-1 所示,程序员可以将多种与 .NET 兼容的语言结合起来开发.NET 应用。多个程序员可以共同参与一个软件项目,每个程序员可以使用自己精通的.NET 支持的编程语言进行编码。

表 1-1 .NET 支持的编程语言

C#	Oberon	Curriculum	Python	J#	Standard ML
APL	Oz	Eiffel	RPG	JScript .NET	Visual Basic .NET
COBOL	Pascal	FORTRAN	Scheme	Mercury	Visual C++ .NET
Component Pascal	Perl	Haskell	Smalltalk	F#	

在了解了.NET 的概念之后,再来看看什么是 ASP.NET。

ASP.NET 是一个统一的 Web 开发模型,它提供了为建立和部署企业级 Web 应用所必需的服务。同时,ASP.NET 是 Microsoft .NET Framework 的一部分,是一种可以在高度分布的 Internet 环境中简化应用程序开发的计算环境。

当编写 ASP.NET 应用程序代码时,可以访问 .NET Framework 中的类。可以使用与公共语言运行库(Common Language Runtime,CLR)兼容的任何语言来编写应用程序代码。使用这些语言,可以开发具有公共语言运行库、类型安全、继承等方面优点的 ASP.NET 应用。所以它具有如下特点:

(1) ASP.NET 是同 Microsoft .NET Framework 集成在一起的,运行在 CLR 运行库环境中。

ASP.NET 建立在 .NET Framework 的编程类之上,它提供了一个 Web 应用程序模型,并且包含使生成 ASP Web 应用程序变得简单的控件集和结构。ASP.NET 包含封装公共 HTML 用户界面元素(如文本框和下拉菜单)的控件集,这些控件在 Web 服务器上运行,并以 HTML 的形式将它们的用户界面推送到浏览器。在服务器上,这些控件公开一个面向对象的编程模型,为 Web 开发人员提供了面向对象编程的丰富性。ASP.NET 还提供了结构服务(如会话状态管理和进程回收),进一步减少了开发人员必须编写的代码量,并提高了应用程序的可靠性。另外,ASP.NET 能让开发人员以服务的形式交付软件。使用 XML Web Service 功能,ASP.NET 开发人员可以编写自己的业务逻辑并使用 ASP.NET 结构,最后通过 SOAP 交付该服务。

(2) ASP.NET 是编译执行的,它支持多种编程语言。同时,它也是面向对象的。

在 ASP.NET 应用开发中,可以使用与 CLR 兼容的任何语言来编写应用程序的代码,如 Microsoft Visual Basic、C#、JScript.NET 和 J# 等编程语言。以 C# 为例,它会经过两个阶段的编译过程:

第一个阶段,编写的 C# 代码首先被 C# 编译器编译成 .NET 的中间语言(Intermediate Language,IL)。实际上,所有 .NET 语言(包括 Visual Basic、C# 等)都会编译成相同的 IL 代码,这也是 .NET 为什么能够实现与语言无关性的关键所在。当页面被第一次请求的时候,第一步的编译过程会自动执行,当然也可以提前执行(将此称为预编译),这个编译的 IL 代码文件称为程序集。

第二个阶段,在这个页面实际执行的时候开始。此时,IL 代码被编译成本机机器代码,将此阶段称为即时编译(Just-In-Time,JIT)。

其实,ASP.NET 应用程序不必在每次请求网页的时候都进行编译,这些中间语言代码在源文件被修改之前只被编译一次。关于代码究竟什么时候编译成 IL 代码,这取决于创建 Web 项目的方式。在 Microsoft Visual Studio 2010 中,如果采用新建 Web 应用程序的方式来创建 Web 项目,那么页面代码在第一次请求的时候才被编译成 IL。无论哪种方式,代码都是在第一次执行时进入编译的第二阶段,即从 IL 到本机机器代码。

(3) ASP.NET 是跨浏览器和跨设备的。

要做到跨任何浏览器运行是所有 Web 开发人员所面临的巨大挑战,同时,它也是衡量一门 Web 编程语言的重要指标。但对于今天的 ASP.NET 程序员来讲,跨浏览器的问题

题似乎变得不那么重要。如果你在开发中完全使用 ASP.NET 自带的 Web 服务器控件,那么这些 Web 服务器控件将会根据客户端的浏览器来自动生成相应的 HTML。这样,不用编些任何其他的额外代码就能够实现跨浏览器支持。

(4) ASP.NET 易于配置与部署。

ASP.NET 的易于配置与部署的特性是任何一个开发平台都不能相比的。微软公司在这方面一直都做得非常好。尤其是在 Windows 7 和 Windows Server 2008 操作系统中自带了 .NET 之后,通过复制程序的方式就能够让程序自由运行。

1.1.4 PHP 编程技术

PHP(Hypertext Preprocessor)是一种通用开源脚本语言。其语法吸收了 C、Java 和 Perl 的特点,利于学习,使用广泛,主要适用于 Web 开发领域。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创的语法,可以比 CGI 或者 Perl 更快速地执行动态网页。在制作动态页面时,PHP 是将程序嵌入到 HTML(标准通用标记语言下的一个应用)文档中去执行,执行效率比完全生成 HTML 标记的 CGI 要高许多。PHP 还可以执行编译后的代码,编译可以达到加密和优化代码运行的目的,使代码运行更快。

PHP 于 1994 年由 Rasmus Lerdorf 创建,最初是 Rasmus Lerdorf 为了维护个人网页而制作的一个简单的用 Perl 语言编写的程序。这些工具程序用来显示 Rasmus Lerdorf 的个人履历以及统计网页流量。后来又用 C 语言重新编写,包括可以访问数据库。他将这些程序和一些表单解释程序整合起来,称为 PHP/FI。PHP/FI 可以和数据库连接,产生简单的动态网页程序。

【实例 1-1】 一个简单的 PHP 程序。

```
<html>
<head>
  <title>php</title>
</head>
<body>
  <?php
```

```
  echo "this is my php code!";
```

```
  ?>
</body>
</html>
```

PHP 编程技术有如下优势:

- 开放源代码。所有的 PHP 源代码事实上都可以得到。
- 免费性。PHP 本身免费且是开源代码。
- 快捷性。程序开发快,运行快,技术本身学习快。
- 嵌入 HTML。因为 PHP 可以被嵌入 HTML 语言,相对于其他语言,PHP 编辑简单,实用性强,更适合初学者。
- 跨平台性强。由于 PHP 是运行在服务器端的脚本,可以运行在 UNIX、Linux、