



全国高等职业教育规划教材

ASP.NET 程序设计 案例教程

第2版

郭力子 华 驰 编著

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>Html 文件的结构</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body bgcolor="#FFFFFF" >
<h2>Html 文件的基本结构</h2>
<b>新浪</b>公司的网址是
<br/>
<a href="http://www.sina.com.cn">新浪公司</a>
</body>
</html>
```

电子课件、源代码下载网址 www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS

全国高等职业教育规划教材

ASP.NET 程序设计 案例教程

第2版

郭力子 华 驰 编著



机械工业出版社

本书从应用的角度出发,系统地介绍了运用 ASP.NET 进行 Web 应用开发的基础知识和基本技能。全书按任务驱动模式编写,共分 10 个学习单元,分别是:了解 ASP.NET 和开发环境、HTML 网页设计基础、C# 语言程序设计基础、常用服务器控件、ASP.NET 程序的格式处理、状态管理与程序配置、使用数据源控件访问数据库、使用 ADO.NET 访问数据库、ASP.NET 程序的文件处理和 ASP.NET 应用程序项目。

本书的 10 个学习单元包含 19 个学习任务,内容循序渐进。各学习任务配备了数量丰富的案例,内容立足于课堂教学和面向应用。各主要学习单元后附有实训和习题供读者实践、练习、巩固与提高。

本书全面兼顾了知识介绍、编程能力培养和实践技能训练,可作为高职高专或其他高校相关专业的 ASP.NET 程序设计课程教材,也可作为相关培训机构或 .NET 开发人员的教材或参考书。

本书配有电子课件和源代码,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 1239258369, 电话: 010-88379739)。

图书在版编目(CIP)数据

ASP.NET 程序设计案例教程/郭力子,华驰编著. —2 版. —北京:机械工业出版社,2015.3

全国高等职业教育规划教材

ISBN 978-7-111-50035-3

I. ①A… II. ①郭… ②华… III. ①网页制作工具-程序设计-高等职业教育-教材 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2015)第 081633 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:鹿 征 责任编辑:鹿 征

责任校对:张艳霞 责任印制:李 洋

北京宝昌彩色印刷有限公司印刷

2015 年 6 月第 2 版·第 1 次

184mm×260mm·18.75 印张·465 千字

0001-3000 册

标准书号:ISBN 978-7-111-50035-3

定价:39.80 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:(010)88379833

机工官网:www.cmpbook.com

读者购书热线:(010)88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面防伪标均为盗版

金书网:www.golden-book.com

全国高等职业教育规划教材计算机专业

编委会成员名单

主 任 周智文

副 主 任 周岳山 林 东 王协瑞 张福强

陶书中 睦碧霞 龚小勇 王 泰

李宏达 赵佩华

委 员 (按姓氏笔画顺序)

马 伟 马林艺 万雅静 万 钢

卫振林 王兴宝 王德年 尹敬齐

史宝会 宁 蒙 安 进 刘本军

刘剑昀 刘新强 刘瑞新 乔芄喆

余先锋 张洪斌 张瑞英 李 强

何万里 杨 莉 杨 云 贺 平

赵国玲 赵增敏 赵海兰 钮文良

胡国胜 秦学礼 贾永江 徐立新

唐乾林 陶 洪 顾正刚 曹 毅

黄能耿 黄崇本 裴有柱

秘 书 长 胡毓坚

出版说明

《国务院关于加快发展现代职业教育的决定》指出：到2020年，形成适应发展需求、产教深度融合、中职高职衔接、职业教育与普通教育相互沟通，体现终身教育理念，具有中国特色、世界水平的现代职业教育体系，推进人才培养模式创新，坚持校企合作、工学结合，强化教学、学习、实训相融合的教育教学活动，推行项目教学、案例教学、工作过程导向教学等教学模式，引导社会力量参与教学过程，共同开发课程和教材等教育资源。机械工业出版社组织全国60余所职业院校（其中大部分是示范性院校和骨干院校）的骨干教师共同策划、编写并出版的“全国高等职业教育规划教材”系列丛书，已历经十余年的积淀和发展，今后将更加紧密结合国家职业教育文件精神，致力于建设符合现代职业教育教学需求的教材体系，打造充分适应现代职业教育教学模式的、体现工学结合特点的新型精品化教材。

“全国高等职业教育规划教材”涵盖计算机、电子和机电三个专业，目前在销教材300余种，其中“十五”“十一五”“十二五”累计获奖教材60余种，更有4种获得国家级精品教材。该系列教材依托于高职高专计算机、电子、机电三个专业编委会，充分体现职业院校教学改革和课程改革的需要，其内容和质量颇受授课教师的认可。

在系列教材策划和编写的过程中，主编院校通过编委会平台充分调研相关院校的专业课程体系，认真讨论课程教学大纲，积极听取相关专家意见，并融合教学中的实践经验，吸收职业教育改革成果，寻求企业合作，针对不同的课程性质采取差异化的编写策略。其中，核心基础课程的教材在保持扎实的理论基础的同时，增加实训和习题以及相关的多媒体配套资源；实践性较强的课程则强调理论与实训紧密结合，采用理实一体的编写模式；涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法，同时重视企业参与，吸纳来自企业的真实案例。此外，根据实际教学的需要对部分课程进行了整合和优化。

归纳起来，本系列教材具有以下特点：

- 1) 围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。
- 2) 合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度，强调专业技术应用能力的训练，适当增加实训环节。
- 3) 符合高职学生的学习特点和认知规律。对基本理论和方法的论述容易理解、清晰简洁，多用图表来表达信息；增加相关技术在生产中的应用实例，引导学生主动学习。
- 4) 教材内容紧随技术和经济的发展而更新，及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念，并积极支持新专业的教材建设。
- 5) 注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合，提高教学服务水平，为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快，加之我们的水平和经验有限，因此在教材的编写和出版过程中难免出现问题和疏漏。我们恳请使用这套教材的师生及时向我们反馈质量信息，以利于我们今后不断提高教材的出版质量，为广大师生提供更多、更适用的教材。

机械工业出版社

前 言

ASP.NET 是由微软公司推出的基于 .NET Framework 的 Web 应用开发平台。作为一种建立在 .NET 公共语言运行库上的 Web 应用编程框架, ASP.NET 为开发内容丰富的、动态的、个性化的 Web 应用程序提供了一种高效的方案。使用 ASP.NET 进行 Web 应用程序的开发, 程序结构更加清晰, 开发流程更加简单, 从而可以提高开发效率, 缩短开发周期。因此, ASP.NET 已成为当今 Web 应用开发的主要平台之一。

本书编写的指导思想是强调实际应用, 注重能力培养。本书内容涵盖了运用 ASP.NET 进行 Web 应用开发的基本知识和技能, 书中示例面向应用, 全面兼顾了知识介绍、编程能力培养和实践技能训练。

本书以任务驱动模式编写, 全部内容按 ASP.NET 程序设计知识体系分 10 个学习单元, 含 19 个学习任务。每个任务的实现都是相关知识的综合案例。这些案例简单明了、针对性强, 大多可以直接用于工程实践。各主要学习单元后配有实训, 很多都是书中案例的扩展, 有助于读者实践; 另外还配有丰富的简答题、填空题和选择题供读者练习。为了帮助读者快速理解, 书中案例前都有关于代码编写的简要分析和步骤, 案例后对重要或易错之处还有说明。读者通过仔细研读这些代码并且经过实践练习, 可以迅速培养和提高独立解决实际编程问题的能力。第 10 学习单元的“ASP.NET 应用程序项目”是课程知识的综合应用, 可帮助读者提高 Web 应用程序项目开发的实际能力。

本书适用于高职高专院校的相关专业, 也可作为 ASP.NET 程序设计培训机构的教材或参考书。对于 ASP.NET 的初学者, 可以按照学习单元安排, 循序渐进地阅读本书, 学习理解并掌握完成各工作任务的实现过程, 从而系统掌握 Web 应用程序的开发技能。对于有一定编程基础和 Web 开发经验的读者, 可以仔细研读示例, 从而提高独立编写 Web 应用程序的能力。

本书的全部示例代码和采用 PowerPoint 制作的电子教案, 可从机械工业出版社教材服务网 <http://www.cmpedu.com> 下载。

在编写过程中, 编者参考了大量相关文献, 在此谨向这些文献的作者深表感谢。

鉴于编者水平有限, 书中难免出现错误和不足之处, 恳请广大读者批评指正。

编 者

目 录

出版说明

前言

学习单元1 了解 ASP.NET 和开发

环境 1

1.1 任务1——了解 ASP.NET 1

1.1.1 任务描述与场景 1

1.1.2 Web 基础知识 1

1.1.3 .NET 和 ASP.NET 2

1.1.4 ASP.NET 的功能与程序组成 3

1.1.5 ASP.NET 的开发环境 5

1.2 任务2——创建第一个 ASP.NET

网页 7

1.2.1 任务描述与场景 7

1.2.2 使用 Visual Studio 2010 建立网页 ... 7

1.2.3 使用 Visual Studio 2010 设计网页 ... 9

1.2.4 为控件添加事件代码 12

1.2.5 预览 ASP.NET 网页 13

1.2.6 发布 ASP.NET 网页 13

1.2.7 aspx 页面元素 14

1.3 任务拓展——使用 IIS 14

1.4 实践与练习 18

1.4.1 实训 18

1.4.2 习题 18

学习单元2 HTML 网页设计基础 19

2.1 任务1——使用 HTML 设计

网页 19

2.1.1 任务描述与场景 19

2.1.2 HTML 简介 19

2.1.3 HTML 语法结构 20

2.1.4 头部标记 21

2.1.5 主体标记 21

2.1.6 格式标记 24

2.1.7 文本标记 27

2.1.8 超链接标记 30

2.1.9 图像标记 32

2.1.10 表格标记 33

2.1.11 表单标记 34

2.1.12 任务实现 36

2.2 任务2——建立网页框架 39

2.2.1 任务描述与场景 39

2.2.2 框架集标记 39

2.2.3 任务实现 39

2.3 任务拓展——使用 JavaScript 41

2.4 实践与练习 46

2.4.1 实训 46

2.4.2 习题 46

学习单元3 C#语言程序设计基础 47

3.1 任务1——用 C#实现交互式

网页 47

3.1.1 任务描述与场景 47

3.1.2 C#的数据类型 47

3.1.3 类型转换 51

3.1.4 变量与常量 52

3.1.5 运算符与表达式 55

3.1.6 C#流程控制 57

3.1.7 函数 63

3.1.8 异常处理 65

3.1.9 任务实现 66

3.2 任务2——用 C#创建公共类 70

3.2.1 任务描述与场景 70

3.2.2 面向对象程序设计特征 70

3.2.3 类的定义 71

3.2.4 类的成员 71

3.2.5 属性、方法与事件 72

3.2.6 对象和类的实例化 73

3.2.7 构造函数和析构函数 74

3.2.8 任务实现 74

3.3 实践与练习 77

3.3.1 实训 77

3.3.2 习题 78

学习单元4 常用服务器控件	80
4.1 任务1——使用服务器控件设计网页	80
4.1.1 任务描述与场景	80
4.1.2 服务器控件概述	80
4.1.3 文本类型控件	86
4.1.4 按钮类控件	87
4.1.5 选择类型控件	88
4.1.6 其他常用标准控件	93
4.1.7 用户控件	99
4.1.8 任务实现	100
4.2 任务2——用验证控件实现网页验证	103
4.2.1 任务描述与场景	103
4.2.2 RequiredFieldValidator 控件	103
4.2.3 CompareValidator 控件	104
4.2.4 RangeValidator 控件	105
4.2.5 RegularExpressionValidator 控件	105
4.2.6 CustomValidator 控件	106
4.2.7 ValidationSummary 控件	106
4.2.8 任务实现	106
4.3 任务拓展——在网页中使用 AJAX 技术	108
4.3.1 ASP.NET 与 AJAX	108
4.3.2 使用 AJAX	108
4.4 实践与练习	110
4.4.1 实训	110
4.4.2 习题	111
学习单元5 ASP.NET 程序的格式处理	113
5.1 任务1——设计网站界面风格	113
5.1.1 任务描述与场景	113
5.1.2 使用 CSS	113
5.1.3 主题	116
5.1.4 母版页	119
5.1.5 任务实现	121
5.2 任务2——实现网站导航功能	124
5.2.1 任务描述与场景	124
5.2.2 XML 文件	125
5.2.3 使用导航控件	125
5.2.4 任务实现	130
5.3 实践与练习	131
5.3.1 实训	131
5.3.2 习题	132
学习单元6 状态管理与程序配置	133
6.1 任务1——用状态管理实现网络聊天室	133
6.1.1 任务描述与场景	133
6.1.2 ASP.NET 的内置对象概述	133
6.1.3 Page 对象	133
6.1.4 Request 对象	136
6.1.5 Response 对象	140
6.1.6 Cookie 对象	143
6.1.7 Session 对象	146
6.1.8 Application 对象	150
6.1.9 Server 对象	152
6.1.10 任务实现	155
6.2 任务2——完成网站程序设置	160
6.2.1 任务描述与场景	160
6.2.2 使用 Web. Config 配置文件	160
6.2.3 使用 Global. asax 配置文件	163
6.2.4 Web 程序身份验证与授权	166
6.2.5 任务实现	168
6.3 实践与练习	171
6.3.1 实训	171
6.3.2 习题	171
学习单元7 使用数据源控件访问数据库	174
7.1 任务1——使用数据源控件和 GridView 等数据控件	174
7.1.1 任务描述与场景	174
7.1.2 数据绑定	174
7.1.3 数据源控件	176
7.1.4 数据控件 GridView	180
7.1.5 数据控件 DetailsView	185
7.1.6 任务实现	185
7.2 任务2——使用 DataList 和 Repeater 数据控件	188

7.2.1 任务描述与场景	188	9.2.3 在 ASP.NET 中输入/输出 Excel 文件	252
7.2.2 数据控件 DataList	189	9.2.4 任务实现	252
7.2.3 数据控件 Repeater	191	9.3 实践与练习	255
7.2.4 任务实现	194	9.3.1 实训	255
7.3 实践与练习	197	9.3.2 习题	255
7.3.1 实训	197		
7.3.2 习题	197	学习单元 10 ASP.NET 应用程序	
学习单元 8 使用 ADO.NET 访问 数据库	199	项目	256
8.1 任务——使用 ADO.NET 访问 数据库	199	10.1 任务 1——ASP.NET 应用程序 项目示例	256
8.1.1 任务描述与场景	199	10.1.1 任务描述与场景	256
8.1.2 ADO.NET 概述	199	10.1.2 校园二手商品交易信息 系统概述	256
8.1.3 ADO.NET 工作方式	201	10.2 系统需求分析	256
8.1.4 Connection 对象	202	10.3 总体设计	257
8.1.5 Command 对象	204	10.3.1 设计目标	257
8.1.6 DataReader 对象	213	10.3.2 项目规划	257
8.1.7 DataAdapter 对象和 DataSet 对象	215	10.3.3 系统业务流程	257
8.1.8 任务实现	227	10.3.4 系统功能结构	257
8.2 实践与练习	231	10.3.5 开发与运行环境	258
8.2.1 实训	231	10.4 概要设计	258
8.2.2 习题	231	10.4.1 数据库设计	258
学习单元 9 ASP.NET 程序的文件 处理	233	10.4.2 文件夹及文件夹模块设计	260
9.1 任务 1——实现网站文件 管理	233	10.4.3 Web.Config 文件配置	261
9.1.1 任务描述与场景	233	10.5 详细设计	261
9.1.2 文件的 I/O 操作	233	10.5.1 公共类设计	261
9.1.3 文件和目录的管理	242	10.5.2 网站前台主要功能模块设计	263
9.1.4 任务实现	247	10.5.3 前台主要页面	268
9.2 任务 2——实现网站数据 交换	249	10.5.4 网站后台主要功能模块设计	285
9.2.1 任务描述与场景	249	10.5.5 后台主要页面	285
9.2.2 在 ASP.NET 中读写 XML 文件	249	10.6 任务 2——ASP.NET 程序设计 项目实践	289
		10.6.1 任务描述与场景	289
		10.6.2 系统设计	289
		参考文献	291

学习单元1 了解 ASP.NET 和开发环境

学习目标

- 理解静态网页与动态网页的概念及其工作原理
- 了解 ASP.NET 的功能与程序组成
- 了解 VS 2010 中 Web 应用的开发环境
- 掌握在 VS 2010 开发环境中 ASP.NET 应用程序开发的一般流程

1.1 任务1——了解 ASP.NET

1.1.1 任务描述与场景

ASP.NET 是微软公司面向互联网时代的软件开发平台 .NET 框架 (.NET Framework) 上的 Web 开发平台, 是目前最流行的 Web 应用程序开发技术之一。ASP.NET 能够提供各种方便的 Web 开发模型, 利用这些模型用户可以快速地开发出 Web 应用程序所需的各种复杂功能。

本任务的要求是了解 Web 的基本知识、ASP.NET 开发环境与 ASP.NET 程序的构成, 为开始学习 ASP.NET Web 程序开发做准备。

1.1.2 Web 基础知识

WWW (World Wide Web) 简称 Web, 是 Internet 提供的一项最基本、应用最广泛的服务。Web 是存储在 Internet 网络服务器中数量巨大的文档的集合。这些文档称为页面, 是一种超文本 (Hypertext) 信息, 可以用于描述超媒体。文本、图形、视频以及音频等多媒体称为超媒体 (Hypermedia)。这些超文本信息在 Internet 上按 HTTP 协议进行传输。

超文本传输协议 (HyperText Transfer Protocol, HTTP) 是在 Internet 中进行信息传送的协议, 是浏览器默认使用的协议。

从浏览器向 Web 服务器发出的访问某一 Web 网页的请求称为 HTTP 请求。Web 服务器收到 HTTP 请求后, 就会按照请求寻找相应的网页。如果找到, 就把网页以超文本标记语言 (HyperText Markup Language, HTML) 代码形式通过 Internet 传回浏览器; 如果找不到, 就发送一个错误信息给浏览器。后面的这些操作就称为 HTTP 响应。

Web 服务器就是一台安装了 Web 服务器软件的计算机, 微软公司的 Web 服务器软件是 IIS (Internet Information Services)。它可以为提出 HTTP 请求的浏览器提供 HTTP 响应。是 Web 中提供服务的一方。称为服务器端。

Internet 网络中运行浏览器程序的计算机, 被用户用来浏览服务器上的可用资源。当用户进行网页浏览时, 由用户的浏览器执行来自服务器的 HTML 代码, 并将其内容显示给用户。此时, 用户使用的计算机是 Web 中接受服务的一方, 称为客户端。服务器端和客户端示意图如图 1-1 所示。

服务器端和客户端的位置也不是一成不变的。如果原来提供服务的服务器端需要接受别的服务器端的服务, 它就转化成了客户端; 原来接受服务的客户端要为别的客户端提供服务, 它就转化成了服务器端。具体到读者自己的计算机: 如果要访问新浪等网站, 它就是客户端; 如

果读者给自己的计算机安装了 WWW 服务器软件，就可以把自己的计算机用作服务器，让其他浏览者通过网络访问自己的计算机上的 Web 程序。读者在学习 ASP.NET 或其他动态网页设计语言时，为了方便地调试和运行程序，往往把自己的计算机既当作服务器端，又当作客户端。

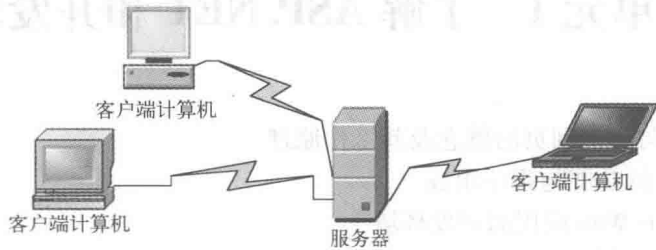


图 1-1 服务器端和客户端示意图

Web 程序的主要呈现方式是 WWW 网页。根据其生成方式，WWW 网页大致可以分为静态和动态网页两种。

所谓静态网页是相对动态网页而言的，是指该网页文件里没有程序代码，只有 HTML 标记（本书将在第 2 章中介绍基本的 HTML 标记）。这种网页一般以扩展名 htm、html、shtml 或 xml 存放。静态网页一经制成，在浏览器中，其内容就固定不变了。

所谓动态网页，是指该网页文件不仅含有 HTML 标记，而且含有一些实现特定功能的程序代码，这些程序代码使得浏览器与服务器之间可以进行交互，即服务器端可以根据客户端的不同请求动态产生网页内容。动态网页的扩展名通常根据所用的程序设计语言的不同而不同。网页能够根据不同的时间、不同的访问者而显示不同的内容。“动态”主要指的是“交互性”，这与动画、Flash 等只是视觉上的所谓动态网页是不同的概念。如常见的论坛、留言板、聊天室、博客、网上购物以及网上银行等通常都是采用动态网页实现的。

一般来说，静态网页较为容易生成。通常采用 Dreamweaver、FrontPage 等软件来完成。而动态网页的制作较为复杂，需要用到 ASP、PHP、JSP 以及本书介绍的 ASP.NET 等专门的动态网页设计语言。

1.1.3 .NET 和 ASP.NET

ASP.NET 是基于微软公司的 .NET 框架的 Web 开发平台，为此有必要先了解一下 .NET 框架（.NET Framework）。.NET 框架是微软公司的一个多语言组件开发和运行环境，其目的是在统一框架下，可以应用多种语言，创建、配置和运行 Windows 应用程序和 Web 应用程序。.NET 框架主要由以下几部分组成，如图 1-2 所示。

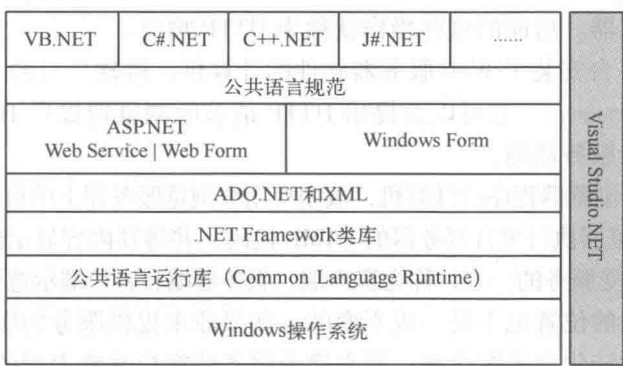


图 1-2 .NET 框架的组成

- 公共语言运行库（Common Language Runtime，CLR）；
- .NET Framework 类库（.NET Framework Class Library）；
- Microsoft ADO.NET；
- ASP.NET；
- VB.NET、C#.NET、C++.NET 等语言运行库。

公共语言运行库（CLR）是.NET框架的运行环境，它负责运行和维护用户编写的任何程序代码，包括内存管理、线程执行、代码执行、代码安全验证、编译及其他系统服务等。符合公共语言规范（Common Language Specification，CLS）的任何语言都可以在公共语言运行库上运行。.NET框架支持的主要语言有 Visual Basic.NET（简称 VB.NET）、C#.NET、C++.NET 和 J#.NET。

.NET 框架工作时，CLR 将所有代码先编译成 MSIL（MicroSoft Intermediate Language，微软中间语言），然后再编译成本机机器语言代码，因此，理论上.NET可以应用在 UNIX、Linux、Mac OS 或其他操作系统上。而且由于 CLR 的存在，使得.NET Framework 消除了异类框架之间的差别，可以让开发人员选择其自己喜欢或擅长的编程语言。

.NET 框架通常由安装程序自动安装，如安装 Visual Studio 2010 软件包时自动安装.NET Framework 4.0。

1.1.4 ASP.NET 的功能与程序组成

1. ASP.NET 应用程序功能

ASP.NET 是建立在 .NET Framework 基础之上的 Web 应用程序开发平台。作为 .NET Framework 的组成部分，可用基于.NET的语言 VB.NET 或 C#.NET 来开发 ASP.NET Web 应用程序。目前利用 ASP.NET 可以方便地开发网上聊天室、留言板、BBS 论坛、电子商务和网上教育、网上社区以及网上办公系统等应用程序。

2. ASP.NET 应用程序的文件夹

为了方便程序开发中的管理和操作，ASP.NET 保留了一些文件夹名称，程序开发人员可以直接使用，也可以在应用程序中根据需要增加任意多个文件和文件夹。ASP.NET 应用程序的保留文件夹名如表 1-1 所示。

表 1-1 ASP.NET 应用程序的保留文件夹名（目录）

文件夹名	说明
App_Browsers	包含 ASP.NET 用于标识个别浏览器并确定其功能的浏览器定义文件（.browser）
App_Data	含应用程序数据文件，包括 MDF 文件、XML 文件和其他数据存储文件。ASP.NET 使用来存储应用程序的本地数据库
App_GlobalResources	包含编译到全局范围程序集当中的资源（.resx 和 .resources 文件）
App_LocalResources	包含与应用程序特定页、用户控件或母版页关联的资源（.resx 和 .resources 文件）
App_Theme	包含用于定义 ASP.NET 网页和控件外观的文件集合（.skin 和 .css 文件及图像文件和其他资源）
App_WebReferences	包含用于在应用程序中使用的 Web 引用的引用协定文件（.wsdl 文件）、XML 架构文件（.xsd 文件）和发现文档文件（.disco 和 .discomap 文件）
Bin	包含已编译程序集（.dll 文件）。这些程序集通常是在应用程序中引用的控件、组件或其他代码。应用程序将自动引用此目录中的代码所表示的任何类

3. ASP.NET 应用程序的文件类型

ASP.NET 使用特定的文件类型。在 ASP.NET 开发中，应用程序可能包含如下类型的一个或多个文件。

- Web 窗体 (.aspx 文件)：为 Web 应用程序提供用户接口。
- 代码隐藏页 (.aspx.vb 和 .aspx.cs 文件)：代码隐藏页是和 Web 窗体页关联在一起的，它包含 Web 窗体的服务器端代码。
- Web 服务 (.asmx 文件)：为其他计算机提供共享应用程序的服务。
- web.config 配置文件：是为 Web 应用程序和 Web 服务器定义了默认设置（如数据库连接、安全设置、状态管理等）的 XML 文件。一个 Web 应用程序可拥有多个 web.config 文件，分别位于不同的文件目录下。
- Global.asax 配置文件：用于处理应用程序级事件的可选文件，位于 ASP.NET 应用程序的根目录下。一个 Web 应用程序只能有一个 Global.asax 配置文件。
- 用户控件文件 (.ascx 文件)：该文件定义可重复使用的自定义用户控件。
- 其他组件：包含其他组件的第三方程序集，如 .dll 文件等。
- 应用程序目录中还可以包含其他资源，如样式表、图像和 XML 文件等。

ASP.NET 应用程序的主要文件类型如表 1-2 所示。

表 1-2 ASP.NET 应用程序的主要文件类型

文 件 名	用途与说明
*.aspx	窗体文件，包含 ASP.NET 程序代码的文件，浏览器可执行此类文件，向服务器提出浏览请求
*.cs 或 *.vb	代码文件，cs 文件是存放 C#.NET 代码的文件，而 vb 文件是存放 VB.NET 代码的文件
*.ascx	用户控件文件，包含了用户自定义的控件，可嵌入在多个 .aspx 文件中
*.resx	资源文件，任何不可执行的数据都是资源，资源可能包含大量窗体中的数据，包括字符串、图像等。把数据保存到资源文件中允许在不用重新编译应用程序的情况下改变数据
style.css	样式表文件，Web 应用程序默认样式表文件
Global.asax	配置文件，ASP.NET 应用程序级别的逻辑和事件处理代码都存放在这个文件中，不处理用户界面的生成
web.config	配置文件，包含公共语言运行库能读取的 Web 应用程序配置设置，以及应用程序能读取的设置

4. ASP.NET Web 窗体基本概念

在 ASP.NET 中，Web 窗体（Web Form）指开发的网页。它是一个交互式的窗体（由页面中的 <form> 元素定义），是服务器端与客户端浏览器之间数据传递的一种结构模式。在 Web Form 中包含了 Html 标记、服务器端控件以及相关的程序。

Web Form 由“可视化组件（Visual Component）”和“用户接口逻辑（User Interface Logic）”组成。其中，前者指包含 Html 标记及服务器控件声明的部分（也就是可在浏览器上看到的部分）；后者则指用于实现服务器和用户交互的代码（采用 C# 或 VB.NET 编写的程序）。ASP.NET 提供两种模式来组织页面元素和代码：一种是单一文件模式；另一种是代码隐藏页模式。两种模式的功能是一样的，可以在两种模式中使用同样的控件和代码，但使用方式有所不同。

（1）单文件页模式

页面的静态文本和代码在同一个 .aspx 文件中，程序代码包含在用 <script runat = "server" > </script> 界定的服务器程序脚本代码块中，并且代码中间可以实现对一些方法和属性以及其他代码的定义。

（2）代码隐藏页模式

把一个 Web 窗体分为一个 .aspx 文件和一个对应的 .aspx.cs 文件，前者是静态文本（界面代码），后者则是一些控制代码（可以选择 C# 或 VB.NET 编写），也称为代码隐藏页，依靠 .aspx 文件顶部的页面设置（CodeFile = "xx.aspx.cs" Inherits = "_xx"）把两个文件联系在一起。

代码隐藏页模式将可视化元素和程序代码分别放在不同的文件中，代码文件的后缀根据使用语言的不同而不同。如果使用 C#，则可视化页面元素文件为 .aspx 文件，程序代码文件为 .cs 文件。这种模式也称为代码分离模式。

1.1.5 ASP.NET 的开发环境

1. Visual Studio.NET

Visual Studio 系列产品是当前最具影响力的集成开发环境，它提供了一整套开发工具，可生成 ASP.NET Web 应用程序、Web 服务应用程序、Windows 应用程序和移动设备应用程序等。能够帮助不同类型的开发人员快速创建各类应用程序。本书使用 Visual Studio 2010（对应 .NET Framework 4.0）进行讲解。

☞说明：

- 需要指出，对于初学者而言，在 Visual Studio 系列产品中，选用自 framework 2.0 发布后的 Visual Studio 2005 及后续版本用于学习并无本质区别。读者可以根据实际情况自行选择。

2. Visual Studio 2010 主要界面

启动 Visual Studio 2010 以后，将呈现 Visual Studio 2010 的起始页，如图 1-3 所示。在进行 ASP.NET 程序开发时，主窗口内根据不同需要可以呈现不同的子窗口，主要包括工具箱、代码编辑器、解决方案资源管理器和属性窗口等，如图 1-4 所示。

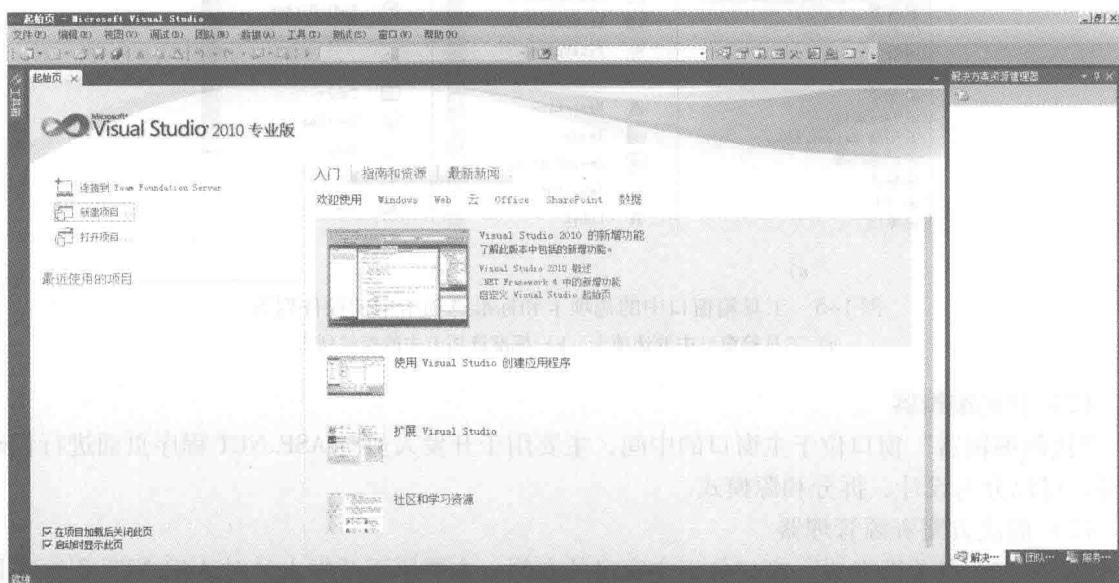


图 1-3 Visual Studio 2010 起始页

(1) 工具箱

“工具箱”是 Visual Studio 的基本窗口，位于 Visual Studio 主窗口的左侧，“工具箱”中包含了 Visual Studio 对 .NET 应用程序所支持的控件。对于不同的应用程序，“工具箱”中所呈现的工具也不同。开发人员可以使用“工具箱”中的控件进行应用程序开发，工具箱窗口中的选项卡和标准选项卡中的控件列表如图 1-5a、b 所示。

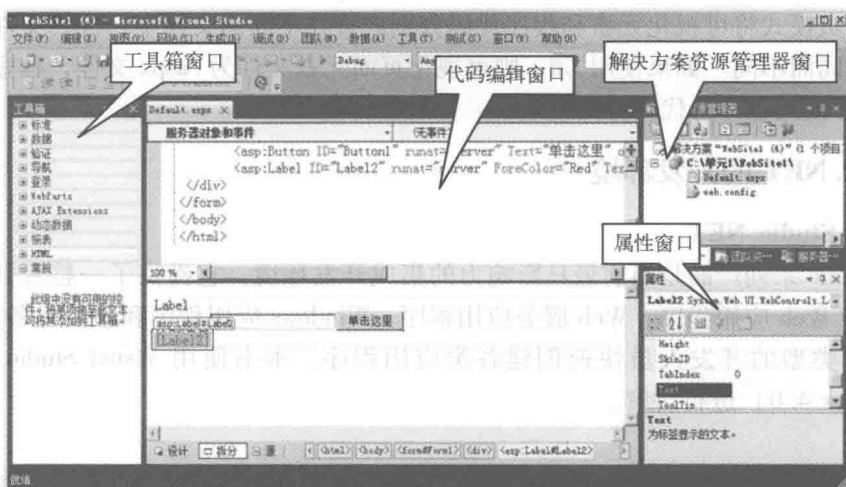


图 1-4 Visual Studio 的主要窗口

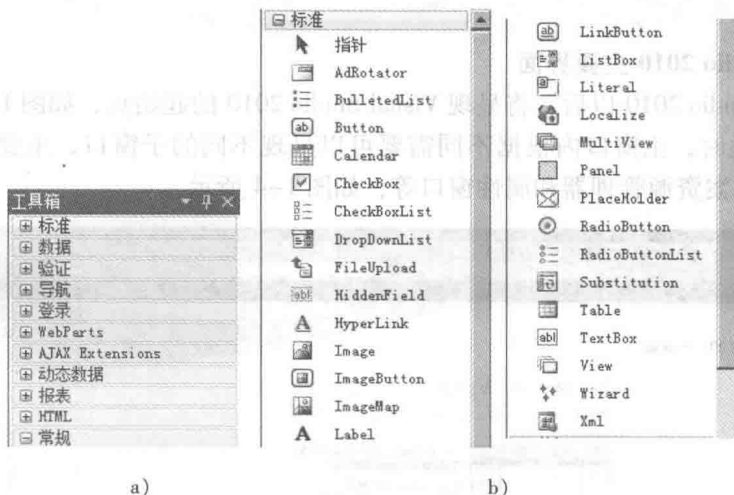


图 1-5 工具箱窗口中的选项卡和标准选项卡中的控件列表

a) 工具箱窗口中的选项卡 b) 标准选项卡中的控件列表

(2) 代码编辑器

“代码编辑器”窗口位于主窗口的中间，主要用于开发人员对 ASP.NET 程序页面进行代码编写，可以分为设计、拆分和源模式。

(3) 解决方案资源管理器

“解决方案资源管理器”窗口位于主窗口的右侧。主要用于开发人员对 ASP.NET 程序项目进行管理、浏览，或向项目中添加新的资源，如文件夹、各类窗体文件和各类资源文件等。

(4) 属性

“属性”窗口位于主窗口的右下侧。为了方便开发人员进行应用程序的开发，Visual Studio 提供了许多控件，每个控件都有自己的属性，通过配置控件的不同属性可以实现复杂的功能。

说明：

如上述的子窗口因关闭后无法显示在 Visual Studio 的主窗口时，可以从主窗口的“视图”菜单下单击相对应的子窗口的菜单项，以再次显示它们。

1.2 任务2——创建第一个 ASP.NET 网页

1.2.1 任务描述与场景

ASP.NET 使用 Visual Studio 集成开发环境实施 Web 程序开发。Visual Studio 界面友好，使用方便，是目前进行 Web 应用开发的主要平台。

本任务的要求是采用 Visual Studio 2010 创建 ASP.NET 网页。通过该过程，帮组读者掌握在 Visual Studio 中创建网页的简要过程与步骤。

1.2.2 使用 Visual Studio 2010 建立网页

Visual Studio 2010 提供了多种程序模板，使开发人员能够快速地进行各种应用程序的开发。以便读者更好地了解其基本原理，在此，我们使用创建空网站的方式来建立一个 ASP.NET 的基本网页。

启动 Visual Studio 2010 应用程序后，选择“文件”→“新建”→“ASP.NET 空网站”，打开“新建网站”对话框，如图 1-6 所示。



图 1-6 “新建网站”对话框

在对话框左边的树形结构中选择“Visual C#”，然后在中间窗口中选择“ASP.NET 空网站”选项，输入网站名称，选择网站所存放的位置，这里输入 c:\单元1\WebSite2，单击“确定”按钮，系统会在右侧的“解决方案资源管理器”窗口创建空网站框架，如图 1-7 所示。

在“解决方案资源管理器”窗口中，用鼠标选中网站名，按右键选择“添加新项”，系统弹出“添加新项”对话框，如图 1-8 所示。在对话框中选择“Web 窗体”，输入网页（Web 窗体）名称，或使用系统给出的默认网页名称 Default.aspx。

注意对话框中右下方的“将代码放在单独的文件中”复选框。如该复选框被选中，表示当前创建的网页采用代码隐藏页模式，即网页由 Default.aspx 文件和一个对应的 Default.aspx.cs 文件组成，前者用于放置和编辑可视化页面元素如 HTML 静态文本等，后者用于编写程序代码。如该选择按钮不被选中，表示当前创建的网页采用单文件页模式，该页面的所有的可视化页面元素和程序代码都位于一个 Default.aspx 文件中。

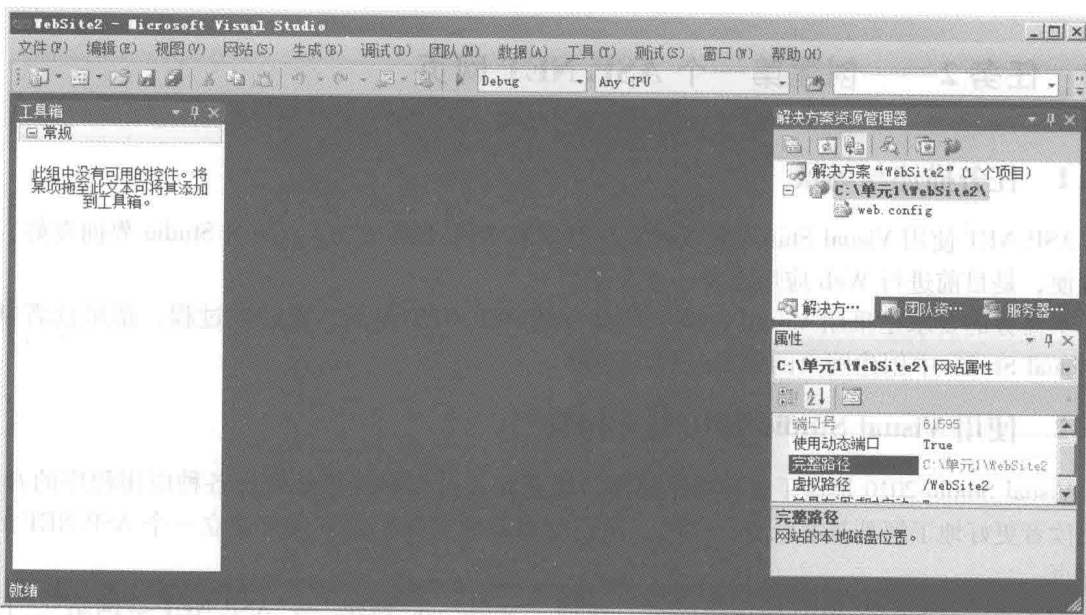


图 1-7 “解决方案资源管理器”窗口

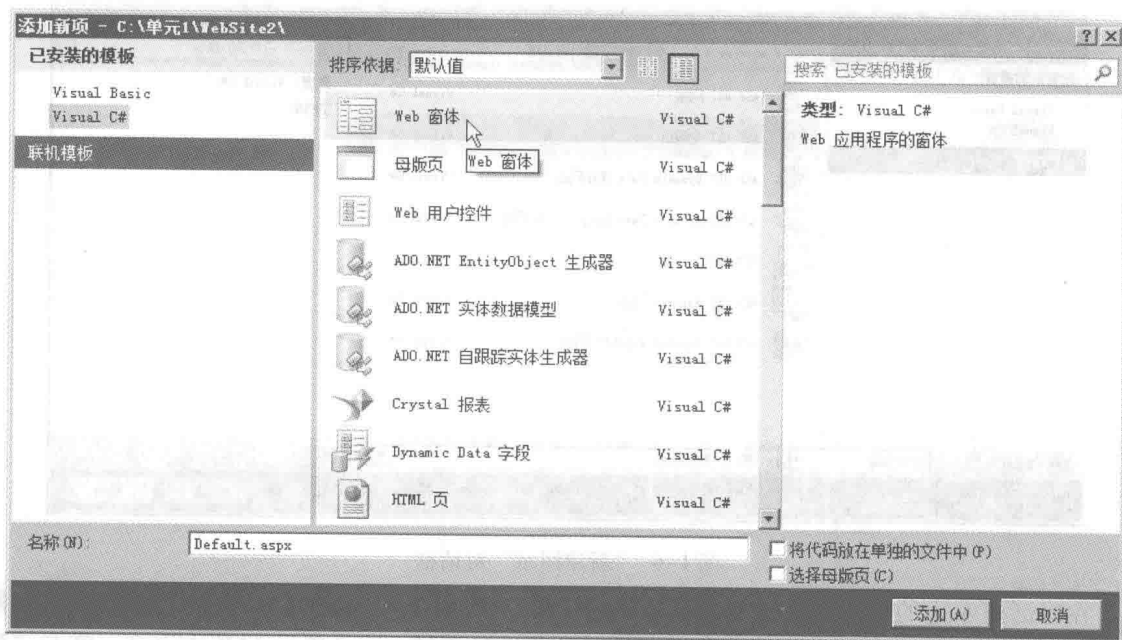


图 1-8 “添加新项”对话框

说明：

- 采用代码隐藏页模式的网页和采用单文件页模式的网页功能相同，运行时没有性能差异。选择何种模式取决于开发人员的喜好。一般来说，对于那些代码不太复杂的网页来说，常采用单文件页模型；而对于代码比较复杂的网页，则常采用代码隐藏页模型。
- 单文件页模型的 aspx 页面适用于较为简单的网页，可以在一个地方看到代码和 html 标记，便于学习。这也是本书后续章节中主要采纳的方法。