



“十二五”职业教育国家规划教材  
经全国职业教育教材审定委员会审定



普通高等教育“十一五”国家级规划教材

# 基于C#的ASP.NET 程序设计 第3版

翁健红 言海燕 主编



 机械工业出版社  
CHINA MACHINE PRESS



配电子课件



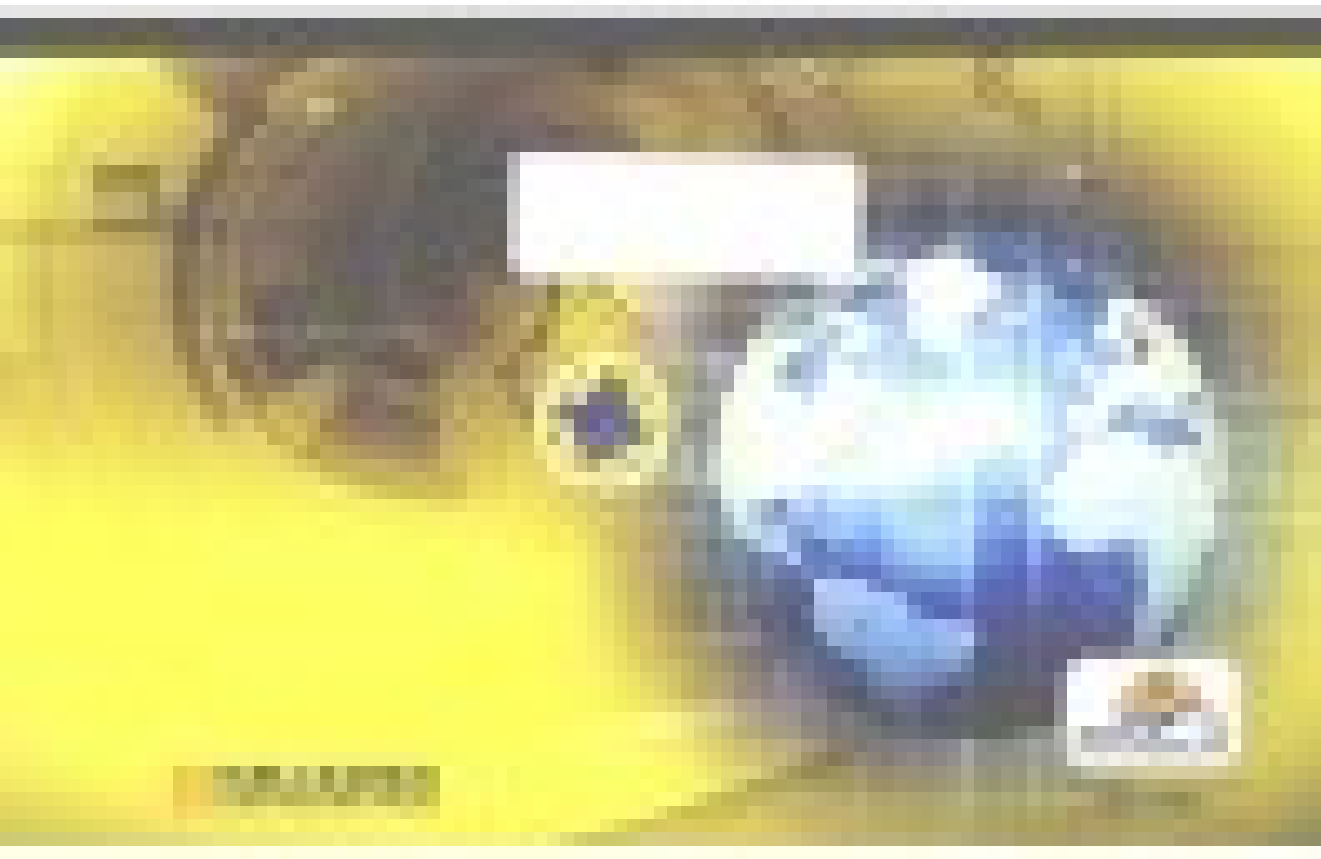
中华人民共和国教育部推荐教材  
普通高等教育计算机专业系列教材



教育部推荐教材 计算机专业系列教材

# 基于C#的ASP.NET 程序设计 第3版

陈维红 陈维红 主编



清华大学出版社





“十二五”职业教育国家规划教材  
经全国职业教育教材审定委员会审定  
普通高等教育“十一五”国家级规划教材

# 基于 C# 的 ASP.NET 程序设计

第 3 版

主 编 翁健红 言海燕

副主编 马绍惠 彭 勇 魏 丽

参 编 孙桂芝 刘小华



机械工业出版社

ASP.NET 是 Web 开发的主流技术之一。本书主要介绍使用 ASP.NET 2.0 进行 Web 应用系统的开发,使用 C#作为 ASP.NET 的开发语言,开发环境为 VS 2005,数据库采用 SQL Server 2005。

本书共 11 章,内容包括 ASP.NET 基础、C#语言基础、服务器端控件、ADO.NET 数据库访问技术、VS.NET 开发会员管理系统、数据窗体设计、内置对象与缓存、网站导航控件与 Web 服务、母版页与主题、AJAX 技术以及设计实例。

本书内容丰富,结构清晰,叙述深入浅出,适合作为高职高专院校计算机及相关专业 Web 应用系统开发的教材,也可作为 ASP.NET 培训教材或从事 ASP.NET 编程和网站开发人员的参考书。

为方便教学,本书配备电子课件、源代码、习题答案等教学资源。凡选用本书作为教材的教师均可登录机械工业出版社教育服务网 [www.cmpedu.com](http://www.cmpedu.com) 免费下载。如有问题请致信 [cmpgaozhi@sina.com](mailto:cmpgaozhi@sina.com),或致电 010-88379375 联系营销人员。

## 图书在版编目 (CIP) 数据

基于 C#的 ASP.NET 程序设计/翁健红,言海燕主编. —3 版.

—北京:机械工业出版社,2015.1

“十二五”职业教育国家规划教材 普通高等教育“十一五”国家级规划教材  
ISBN 978-7-111-48891-0

I. ①基… II. ①翁… ②言… III. ①网页制作工具—程序设计—高等学校—教材 ②C 语言—程序设计—高等学校—教材 IV. ①TP393.092 ②TP312

中国版本图书馆 CIP 数据核字 (2014) 第 293299 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:王玉鑫 责任编辑:王玉鑫 陈瑞文

责任校对:赵蕊 责任印制:李洋

封面设计:马精明

北京瑞德印刷有限公司印刷 (三河市胜利装订厂装订)

2015 年 1 月第 3 版第 1 次印刷

184mm×260mm·14 印张·322 千字

0001—3000 册

标准书号:ISBN 978-7-111-48891-0

定价:30.00 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

服务咨询热线:010-88379833

读者购书热线:010-88379469

封面无防伪标均为盗版

网络服务

机工官网: [www.cmpbook.com](http://www.cmpbook.com)

机工官博: [weibo.com/cmp1952](http://weibo.com/cmp1952)

教育服务网: [www.cmpedu.com](http://www.cmpedu.com)

金书网: [www.golden-book.com](http://www.golden-book.com)

## 第3版前言

《基于 C#的 ASP.NET 程序设计》一书自 2007 年的第 1 版、2010 年的第 2 版出版以来，被许多高职高专院校选用为教材，受到广大同行的肯定和一致好评，也积累了一些有价值的反馈意见。作者在全面总结前两版成功经验的基础上，根据当前技术发展和教材使用后的反馈信息，对全书进行了修订，主要变动有：由于 AJAX 技术在网站开发中得到了广泛应用，因此增加了相应章节的内容（第 10 章）；为书中实例添加了操作步骤，便于上机练习；删除了一些不常用的内容或例子；增加了 GridView 控件的代码编程。

本书共 11 章，使用 C#作为 ASP.NET 的开发语言，以通俗、简明的语言深入浅出地讲解了用 ASP.NET 2.0 进行 Web 应用系统开发的方法。

本次修订，更加突出以下特色：

### 1. 贯彻够用为度。

在知识点上，不求多和深。根据作者对 ASP.NET 的透彻了解和对市场需求的广泛调研，借鉴现有的各种 IT 教学体系，严格把握“够用、实用”原则，以免学生困难而却步，因多而分心。

### 2. 与案例结合，强调项目、模块的开发。

本书介绍了“会员管理系统”“在线考试系统”与“新闻发布系统”3 个精选的案例。与一般教材把案例放在最后章节不同，本书强调案例为内容服务，在学生有足够知识的情况下，及时引入案例，这样既强化了所学知识，又提高了学生兴趣。

在案例的组织上，引入目标—编程的方式，即在把案例分解为模块的基础上，对每个模块先以“模块设计”说明目标，然后讲解实现这个目标的步骤。这就避免了将案例的讲解变成代码的堆积，使学生无论学习还是上机练习都可以带着明确的目标。

### 3. 内容通俗易懂，没有抽象的说教，而是通过大量具体的实例来阐明原理。

本书由翁健红、言海燕主编，马绍惠、彭勇、魏丽、孙桂芝、刘小华参与了编写。由于时间仓促和作者水平有限，书中不足与疏漏之处在所难免，敬请广大读者批评指正。

编 者

# 目 录

## 第 3 版前言

|                                |    |                                                 |    |
|--------------------------------|----|-------------------------------------------------|----|
| 第 1 章 ASP.NET 基础 .....         | 1  | 3.1.2 Button 控件 .....                           | 30 |
| 1.1 Web 基础知识 .....             | 1  | 3.1.3 TextBox 控件 .....                          | 31 |
| 1.1.1 Web 服务器 .....            | 1  | 3.1.4 RadioButton 控件 .....                      | 32 |
| 1.1.2 静态网页 .....               | 1  | 3.1.5 RadioButtonList 控件 .....                  | 33 |
| 1.1.3 动态网页 .....               | 2  | 3.1.6 DropDownList 控件 .....                     | 35 |
| 1.2 ASP.NET 概述 .....           | 3  | 3.1.7 ListBox 控件 .....                          | 36 |
| 1.3 配置 ASP.NET 的运行环境 .....     | 3  | 3.1.8 Image 控件 .....                            | 38 |
| 1.3.1 ASP.NET 的运行环境 .....      | 3  | 3.1.9 HyperLink 控件 .....                        | 38 |
| 1.3.2 安装 IIS .....             | 4  | 3.1.10 ImageButton 控件 .....                     | 38 |
| 1.3.3 安装 .NET Framework .....  | 4  | 3.1.11 CheckBox 控件 .....                        | 39 |
| 1.3.4 测试 ASP.NET 环境 .....      | 4  | 3.1.12 CheckBoxList 控件 .....                    | 40 |
| 1.4 Visual Studio 集成开发环境 ..... | 5  | 3.1.13 Calendar 控件 .....                        | 41 |
| 1.5 第一个 ASP.NET 程序 .....       | 6  | 3.1.14 Placeholder 控件 .....                     | 43 |
| 1.5.1 代码分离模式 .....             | 6  | 3.1.15 Table 控件 .....                           | 43 |
| 1.5.2 单一文件模式 .....             | 10 | 3.2 数据验证控件 .....                                | 45 |
| 1.6 ASP.NET 页面的结构 .....        | 11 | 3.2.1 RequiredFieldValidator 控件 .....           | 45 |
| 1.7 ASP.NET 应用程序中的特殊文件夹 .....  | 12 | 3.2.2 CompareValidator 控件 .....                 | 46 |
| 1.8 ASP.NET 页面的生命周期事件 .....    | 13 | 3.2.3 RangeValidator 控件 .....                   | 48 |
| 1.9 本书实例采用的数据库 .....           | 15 | 3.2.4 RegularExpressionValidator 控件 ..          | 49 |
| 习题 .....                       | 15 | 3.2.5 ValidationSummary 控件 .....                | 51 |
| 第 2 章 C#语言基础 .....             | 16 | 3.3 HTML 控件简介 .....                             | 52 |
| 2.1 C#语言简介 .....               | 16 | 习题 .....                                        | 52 |
| 2.2 变量 .....                   | 16 | 第 4 章 ADO.NET 数据库访问技术 .....                     | 53 |
| 2.3 数据类型 .....                 | 17 | 4.1 ADO.NET 基本概念 .....                          | 53 |
| 2.4 运算符 .....                  | 19 | 4.2 ADO.NET 对象模型 .....                          | 53 |
| 2.5 程序流程控制 .....               | 22 | 4.3 SqlConnection 对象 .....                      | 55 |
| 2.5.1 条件语句 .....               | 22 | 4.4 Command 对象 .....                            | 56 |
| 2.5.2 循环语句 .....               | 24 | 4.4.1 Command 对象的属性和方法 .....                    | 56 |
| 2.5.3 跳转语句 .....               | 26 | 4.4.2 增加、删除、修改数据 .....                          | 57 |
| 2.6 方法 .....                   | 27 | 4.4.3 执行带参数的 Command .....                      | 58 |
| 习题 .....                       | 28 | 4.4.4 ExecuteScalar 方法 .....                    | 60 |
| 第 3 章 服务器端控件 .....             | 29 | 4.5 DataReader 对象 .....                         | 60 |
| 3.1 常用服务器端控件 .....             | 29 | 4.6 DataSet、DataTable 和<br>DataAdapter 对象 ..... | 62 |
| 3.1.1 Label 控件 .....           | 30 | 4.6.1 DataSet 对象 .....                          | 62 |

|              |                              |            |
|--------------|------------------------------|------------|
| 4.6.2        | DataTable 对象 .....           | 64         |
| 4.6.3        | DataAdapter 对象 .....         | 66         |
| 4.7          | DataView 对象 .....            | 69         |
| 4.8          | 使用存储过程 .....                 | 70         |
| 4.8.1        | 执行不带参数的存储过程 .....            | 70         |
| 4.8.2        | 执行带参数的存储过程 .....             | 71         |
|              | 习题 .....                     | 73         |
| <b>第 5 章</b> | <b>VS.NET 开发会员管理系统 .....</b> | <b>74</b>  |
| 5.1          | 项目基础 .....                   | 74         |
| 5.1.1        | VS.NET 中的表格操作 .....          | 74         |
| 5.1.2        | 程序排错与调试 .....                | 75         |
| 5.1.3        | Session 对象 .....             | 78         |
| 5.1.4        | 编程规范 .....                   | 80         |
| 5.1.5        | Web.config 配置文件 .....        | 82         |
| 5.2          | 会员管理系统 .....                 | 86         |
| 5.2.1        | 系统分析与设计 .....                | 86         |
| 5.2.2        | 系统的运行界面 .....                | 86         |
| 5.2.3        | 配置 Web.config 文件 .....       | 88         |
| 5.2.4        | 登录页面 .....                   | 88         |
| 5.2.5        | 主页面 .....                    | 90         |
| 5.2.6        | 注册页面 .....                   | 91         |
| 5.2.7        | 显示个人信息页面 .....               | 94         |
| 5.2.8        | 修改个人信息页面 .....               | 95         |
| 5.2.9        | 取回密码 1 页面 .....              | 98         |
| 5.2.10       | 取回密码 2 页面 .....              | 100        |
| 5.3          | 发布网站 .....                   | 101        |
|              | 习题 .....                     | 104        |
| <b>第 6 章</b> | <b>数据窗体设计 .....</b>          | <b>105</b> |
| 6.1          | 数据绑定简介 .....                 | 105        |
| 6.2          | Repeater 控件 .....            | 106        |
| 6.3          | DataList 控件 .....            | 108        |
| 6.4          | 数据源控件 .....                  | 111        |
| 6.5          | GridView 控件 .....            | 114        |
| 6.5.1        | GridView 控件简介 .....          | 114        |
| 6.5.2        | GridView 控件的常用属性 .....       | 114        |
| 6.5.3        | GridView 控件的数据绑定列 .....      | 117        |
| 6.5.4        | GridView 控件的数据显示 .....       | 117        |
| 6.5.5        | GridView 控件的数据编辑 .....       | 120        |
| 6.5.6        | GridView 控件中使用下拉列表 .....     | 122        |

|              |                             |            |
|--------------|-----------------------------|------------|
| 6.5.7        | HyperLinkField 列 .....      | 123        |
| 6.5.8        | GridView 控件的编程 .....        | 125        |
| 6.6          | FormView 控件 .....           | 129        |
|              | 习题 .....                    | 132        |
| <b>第 7 章</b> | <b>内置对象与缓存 .....</b>        | <b>133</b> |
| 7.1          | 内置对象 .....                  | 133        |
| 7.1.1        | Application 对象 .....        | 133        |
| 7.1.2        | Request 对象 .....            | 134        |
| 7.1.3        | Response 对象 .....           | 136        |
| 7.1.4        | Server 对象 .....             | 137        |
| 7.2          | Cookie 对象 .....             | 139        |
| 7.2.1        | Cookie 对象的特点 .....          | 139        |
| 7.2.2        | Cookie 对象的读写 .....          | 140        |
| 7.2.3        | Cookie 对象的生存期 .....         | 140        |
| 7.2.4        | Cookie 对象验证的安全性 .....       | 140        |
| 7.2.5        | Cookie 对象应用举例 .....         | 140        |
| 7.3          | Global.asax 文件 .....        | 142        |
| 7.4          | ASP.NET 中的缓存 .....          | 144        |
| 7.4.1        | 输出型缓存 .....                 | 144        |
| 7.4.2        | 片段型缓存 .....                 | 145        |
| 7.4.3        | 数据型缓存 .....                 | 147        |
|              | 习题 .....                    | 148        |
| <b>第 8 章</b> | <b>网站导航控件与 Web 服务 .....</b> | <b>149</b> |
| 8.1          | 网站导航控件 .....                | 149        |
| 8.1.1        | 站点地图 .....                  | 149        |
| 8.1.2        | SiteMapDataSource 控件 .....  | 150        |
| 8.1.3        | Menu 控件 .....               | 151        |
| 8.1.4        | SiteMapPath 控件 .....        | 152        |
| 8.2          | Web 服务 .....                | 153        |
| 8.2.1        | Web 服务简介 .....              | 153        |
| 8.2.2        | Web 服务的作用 .....             | 154        |
| 8.2.3        | Web 服务术语 .....              | 154        |
| 8.2.4        | Web 服务应用实例 .....            | 155        |
|              | 习题 .....                    | 159        |
| <b>第九章</b>   | <b>母版页与主题 .....</b>         | <b>160</b> |
| 9.1          | 母版页 .....                   | 160        |
| 9.1.1        | 母版页概述 .....                 | 160        |
| 9.1.2        | 母版页应用实例 .....               | 160        |



|                                         |            |
|-----------------------------------------|------------|
| 9.1.3 将已建成的网页<br>嵌入母版页中 .....           | 165        |
| 9.1.4 母版页的嵌套 .....                      | 165        |
| 9.1.5 在内容页中访问母版页中的内容 .....              | 165        |
| 9.2 主题与外观 .....                         | 166        |
| 9.2.1 主题与外观概述 .....                     | 166        |
| 9.2.2 应用实例 .....                        | 167        |
| 9.2.3 将主题文件应用于<br>整个应用项目 .....          | 170        |
| 9.2.4 主题应用的规则和优先级 .....                 | 170        |
| 9.3 用户控件 .....                          | 171        |
| 9.3.1 创建用户控件 .....                      | 171        |
| 9.3.2 使用用户控件 .....                      | 172        |
| 习题 .....                                | 173        |
| <b>第 10 章 AJAX 技术 .....</b>             | <b>174</b> |
| 10.1 AJAX 简介 .....                      | 174        |
| 10.2 ASP.NET AJAX 简介 .....              | 175        |
| 10.3 AJAX Extension 安装 .....            | 175        |
| 10.4 ASP.NET AJAX 常用控件 .....            | 176        |
| 10.4.1 ScriptManager 控件 .....           | 176        |
| 10.4.2 UpdatePanel 控件 .....             | 176        |
| 10.4.3 Timer 控件 .....                   | 179        |
| 10.4.4 ScriptManagerProxy 控件 .....      | 179        |
| 10.5 ASP.NET AJAX 应用实例 .....            | 179        |
| 10.5.1 ASP.NET AJAX 实现登录 .....          | 179        |
| 10.5.2 ASP.NET AJAX 实现下拉<br>列表联动 .....  | 181        |
| 10.5.3 ASP.NET AJAX 实现信息<br>的即时刷新 ..... | 182        |
| 10.6 jQuery 的 AJAX 技术 .....             | 183        |
| 习题 .....                                | 185        |

|                            |            |
|----------------------------|------------|
| <b>第 11 章 设计实例 .....</b>   | <b>186</b> |
| 11.1 在线考试系统 .....          | 186        |
| 11.1.1 系统分析与设计 .....       | 186        |
| 11.1.2 系统的运行界面 .....       | 187        |
| 11.1.3 Web.Config 文件 ..... | 188        |
| 11.1.4 数据库操作的公共类 .....     | 188        |
| 11.1.5 考生登录页面 .....        | 191        |
| 11.1.6 考试页面 .....          | 192        |
| 11.1.7 成绩显示页面 .....        | 195        |
| 11.2 新闻发布系统 .....          | 196        |
| 11.2.1 系统分析与设计 .....       | 196        |
| 11.2.2 系统的运行界面 .....       | 197        |
| 11.2.3 浏览新闻页面 .....        | 198        |
| 11.2.4 显示新闻页面 .....        | 199        |
| 11.2.5 添加新闻页面 .....        | 200        |
| 11.2.6 新闻列表页面 .....        | 202        |
| 11.2.7 管理新闻页面 .....        | 207        |
| 11.2.8 修改新闻页面 .....        | 211        |
| 习题 .....                   | 212        |
| <b>附录 参考实验 .....</b>       | <b>213</b> |
| 实验 1 ASP.NET 开发环境 .....    | 213        |
| 实验 2 C# 语言基础 .....         | 213        |
| 实验 3 Web Server 控件 .....   | 213        |
| 实验 4 数据验证控件 .....          | 214        |
| 实验 5 ADO.NET 数据库访问技术 ..... | 214        |
| 实验 6 开发会员管理系统 .....        | 215        |
| 实验 7 数据窗体设计 .....          | 215        |
| 实验 8 内置对象与应用程序配置 .....     | 215        |
| 实验 9 网站导航 .....            | 216        |
| 实验 10 主题与外观 .....          | 216        |
| <b>参考文献 .....</b>          | <b>217</b> |

# 第1章

## ASP.NET 基础



### 本章目标

- 了解 Web 基础知识。
- 掌握配置 ASP.NET 的运行环境。
- 了解 ASP.NET 页面的结构。
- 了解 ASP.NET 页面的事件。

## 1.1 Web 基础知识

超文本传输协议 (Hypertext Transfer Protocol, HTTP) 是在 Internet 中进行信息传送的、被浏览器默认使用的协议。例如, 当用户在浏览器的地址栏中输入 “www.163.com” 的时候, 浏览器会自动使用 HTTP 来搜索 <http://www.163.com> 网站的首页。

从浏览器向 Web 服务器发出的搜索某个 Web 网页的请求称为 HTTP 请求。当 Web 服务器收到这个请求之后, 就会按照请求的要求, 寻找相应的网页。如果可以找到这个网页, 那么就把网页的 HTML 代码通过网络传回浏览器; 如果没有找到这个网页, 那么就发送一个错误信息给发出 HTTP 请求的浏览器。后面的这些操作称为 HTTP 响应。

### 1.1.1 Web 服务器

当提到 Web 服务器的时候, 很多人都会认为这是一台物理的机器。但实际上, Web 服务器是一种软件, 可以管理各种 Web 文件, 并为提出 HTTP 请求的浏览器提供 HTTP 响应。大多数情况下, Web 服务器和浏览器处于不同的计算机, 但是它们也可以并存在同一计算机上。比较常见的 Web 服务器有 Apache 和 IIS。IIS 是由微软公司的 Windows 操作系统提供的, 由于 ASP.NET 只能在 IIS 上运行, 因此在后面的章节中主要介绍 IIS。

### 1.1.2 静态网页

WWW 通过超链接 (Hyper Link) 技术在位于不同位置的文件之间建立连接, 从而可以为用户提供一种交叉式而非线形式的访问方式。借助于这种更符合思维习惯的访问方式, 人们可以十分便捷地访问各种资源 (如文本信息、多媒体信息等)。HTML (Hyper Text Markup Language) 是一种超文本标记语言, 用于声明信息 (如文本、图像等) 的结构、格式和标识超链接等。在文本中嵌入适当的 HTML 标记后所得到的文件称为 HTML

文档。静态网页就是用纯 HTML 代码编写的网页。这些网页的代码是用一些编辑器输入的，或者是用一些网页设计程序生成的，保存为.html 或.htm 文件的形式。设计完成后，无论是哪个用户访问这个网页，在什么时候访问这个网页，以何种方式进入这个网页，它的显示都不会发生任何变化。

**【例 1-1】**下面是一个静态网页的例子，用于显示一个红色的“你好”字符串，运行结果如图 1-1 所示。

**Hello.html**（程序前粗体表示文件名，以后不再一一介绍）



图 1-1 一个简单的静态网页

```
<html>
<body>
<font color=red>你好</font>
</body>
</html>
```

这是一个最简单的 HTML 静态网页，它的目的是显示红色的“你好”字符串。只要这个文件存在，那么任何用户、在任何时间、以任何方式进入这个网页，都将显示同样的结果。

采用静态网页会导致很大的局限性。仅由 HTML 页面构成的 Web 应用程序的内容是静止的，它不会对用户的动作做出动态响应。如果希望为用户显示一些个性化的信息，使用静态网页就无法达到目的。例如，如果当前的时间是新年的开始，就在网页的最上面显示一个“新年好！”的信息；如果当前的时间是圣诞节，就在同样的位置显示一个“圣诞快乐！”的信息，则使用静态网页无法达到上述目的。

### 1.1.3 动态网页

要对用户请求做出动态响应，就要使用动态网页。动态网页可以为不同的用户提供个性化的服务，而为了实现这种动态性，就需要进行程序设计。

Web 应用程序要么在 Web 服务器（作为服务器端脚本时）上执行，要么在 Web 浏览器（作为客户端脚本时）上执行。它们使用户能够通过 Internet 和企业网共享、访问信息。另外，Web 应用程序支持在线商务事务，也就是一般所说的电子商务。

客户端脚本使设计者能开发动态响应用户输入的 Web 页面，而无需和 Web 服务器进行交互。例如，Web 应用程序要求用户输入用户名和密码方可显示主页，要检查用户名和密码部分是否为空白，可以编写运行于客户计算机上的客户端脚本。客户端脚本有助于减少网络流量，因为它不需要和 Web 服务器进行交互作用就可动态响应用户的输入。脚本语言（如 VBScript 和 JavaScript）被用于编写客户端脚本。

服务器端脚本为用户提供动态内容，这些内容是基于远程存储信息（如后台数据库）的，包括用服务器端脚本语言 ASP（Active Server Pages）和 JSP（Java Server Pages）所写的代码。服务器端脚本要在 Web 服务器上执行，当浏览器请求包含服务器端脚本和 HTML 页面时，收到请求的 Web 服务器先处理脚本然后将结果发送给浏览器，这样任何来自 Web 服务器的数据都会用服务器端脚本进行处理。例如，用

Web 页面显示 Web 站点所在系统当前的时间,就需执行服务器端脚本。如果使用了客户端脚本,那么每个请求包含该脚本的文件的浏览器都会显示浏览器所在系统的当前时间。

## 1.2 ASP.NET 概述

ASP.NET (Active Server Pages.NET) 是 Microsoft .NET Framework 中一套用于生成 Web 应用程序和 XML Web services 的技术。ASP.NET 页面是在服务器上执行并生成发送到桌面或移动浏览器的标记(如 HTML、WML 或 XML)。ASP.NET 页面是使用一种已编译的、由事件驱动的编程模型,这种模型可以提高性能并支持将应用程序逻辑同用户界面相隔离。ASP.NET 页面和使用 ASP.NET 创建的 XML Web services 文件包含用 Visual Basic、C# 或任何 .NET 兼容语言编写的服务器端(而不是客户端)逻辑。Web 应用程序和 XML Web services 利用了公共语言运行库的功能,如类型安全、继承、语言互操作、版本控制和集成安全性等。

ASP.NET 是作为 .NET 框架体系结构的一部分推出的。2000 年 ASP.NET 1.0 正式发布,2003 年 ASP.NET 升级为 1.1 版本。ASP.NET 1.1 发布后更加激发了 Web 应用程序开发人员对 ASP.NET 的兴趣,于是在 2005 年 11 月微软公司又发布了 ASP.NET 2.0。ASP.NET 2.0 的发布是 .NET 技术走向成熟的标志。ASP.NET 2.0 技术增加了大量方便、实用的新特性,是一种建立在公共语言运行库上的编程框架,可用于在服务器上开发功能强大的 Web 应用程序。ASP.NET 2.0 不但执行效率大幅度提高,而且对代码的控制也做得更好,同时还支持 Web Controls 功能和多种语言,以高安全性、易管理性和高扩展性等特点著称。

## 1.3 配置 ASP.NET 的运行环境

### 1.3.1 ASP.NET 的运行环境

#### 1. 软件环境

(1) 操作系统: Windows 2000 Professional/Server、Windows XP Professional 或已安装 Service Pack 的 Windows NT 4.0。

(2) 服务软件: Internet Information Services (IIS) 5.0、.NET Framework、MDAC (Microsoft Data Access Components)。此外,如果有 Visual Studio.NET 套件,则只需在 IIS 5.0 基础上安装此套件即可。

(3) WWW 客户: Internet Explorer 5.5 或以上版本。

#### 2. 硬件环境

对运行 ASP.NET 的计算机而言,硬盘及内存越大越好,其中内存最好在 256MB 以上。



## 1.3.2 安装 IIS

如果操作系统是 Windows 2000/2003 Server, 则 IIS 5.0/6.0 是默认安装的。如果是其他 Windows 操作系统, 则需要用户手动安装 IIS。下面以 Windows XP 为例, 说明 IIS 的安装步骤。

- (1) 依次打开“控制面板”→“添加/删除程序”→“添加/删除 Windows 组件”, 勾选“Internet 信息服务 (IIS)”复选框, 如图 1-2 所示, 然后单击“详细信息”按钮。
- (2) 在打开的“Internet 信息服务 (IIS)”对话框中勾选所有复选框, 如图 1-3 所示。

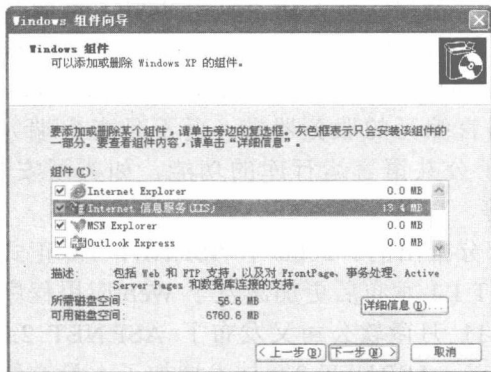


图 1-2 勾选“Internet 信息服务 (IIS)”复选框

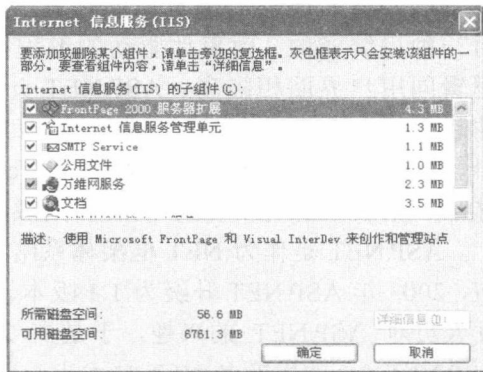


图 1-3 “Internet 信息服务 (IIS)”对话框

(3) 单击“确定”按钮后, 再单击“下一步”按钮。在光驱中放入 WindowsXP 的安装盘, 安装完毕后, 可以测试一下安装是否成功, 在浏览器的地址栏中输入“http://localhost”, 如果安装成功, 则会出现欢迎画面, 如图 1-4 所示。

## 1.3.3 安装 .NET Framework

安装完 IIS 以后, 为了支持 ASP.NET 脚本, 还必须安装 .NET Framework 2.0, 可以在微软的网站下载。需要注意, 一定要先安装 IIS, 再安装 .NET Framework。

## 1.3.4 测试 ASP.NET 环境

安装了 .NET Framework, 系统就具备了运行 ASP.NET 的环境, 可以通过运行一个简单的 ASP.NET 程序来验证。

**【例 1-2】** 建立一个简单的 ASP.NET 程序, 在记事本中输入下面程序:

**first.aspx**

```
<% @ Page Language="C#" %>
<%
```

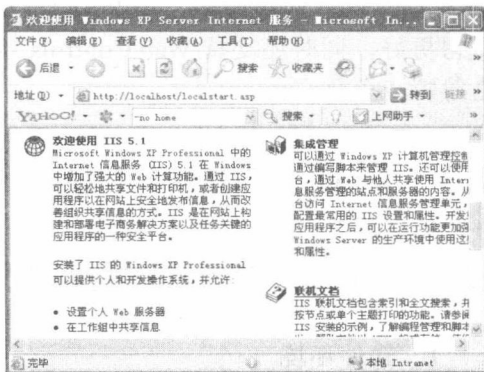


图 1-4 欢迎画面

```
//下面语句输出"您好！"
Response.Write("您好！");
%>
```

程序说明：

- 1) ASP.NET 文件的后缀名为 aspx。
  - 2) C# 对大小写是敏感的。
  - 3) C# 中单行注释用 “//” 标注，注释内容到本行结束为止。多行注释用 /\*...\*/ 把注释内容括起来。
  - 4) C# 用分号 (;) 作为分隔符来终止每条语句。
- 把文件命名为 first.aspx，并保存到 C:\Inetpub\wwwroot 文件夹中，然后在浏览器的地址栏中输入 “http://localhost/first.aspx”，运行结果如图 1-5 所示。此时说明运行环境已配置成功。



图 1-5 运行结果

注意：

- 1) 文件的保存位置要根据系统 Inetpub 目录来决定。例如，如果 Inetpub 目录在 D 盘根目录下，则应存放到 D:\Inetpub\wwwroot 文件夹中。
- 2) localhost 指本机，等同于 127.0.0.1，即网址也可为：http://127.0.0.1/first.aspx。
- 3) 物理磁盘上的 wwwroot 目录应对应网站的主目录，这里，http://127.0.0.1 对应 C:\Inetpub\wwwroot 文件夹。

## 1.4 Visual Studio 集成开发环境

Visual Studio 是 Microsoft 推出的功能强大的集成开发环境，能与 .NET 技术紧密结合，支持建立任意类型的 .NET 组件或应用程序。使用 Visual Studio，可以用与 Microsoft 兼容的编程语言来建立应用程序，还可以创建 Windows Form 与 XML Web 服务、.NET 组件、移动应用程序、ASP.NET 应用程序等。

使用 Visual Studio.NET 的好处是它提供了下列能够使应用程序开发更快速、简易及可靠的工具：

- 1) 可视化的网页设计器。它能够让用户以拖放方式生成控件，并提供具备语法检查功能的 HTML（代码）视图画面。
- 2) 智能型的代码编辑器。它具备命令语句完成、语法检查及其他的智能感知功能。
- 3) 集成的编译与调试能力。
- 4) 项目管理能力。它能够替用户生成与管理应用程序文件，并将文件部署至本机或远程的服务器。

Visual Studio 的发展经历了许多版本，常用的有 Visual Studio 2005、Visual Studio 2008、Visual Studio 2010 等，目前最新的版本为 Visual Studio 2013。对于一般的 ASP.NET 项目的开发，用 Visual Studio 2005 开发环境已足够，并且 Visual Studio 各版本在操作上差异不大。本书例子采用 Visual Studio 2005 开发，Visual Studio 2005 系统启动后的界面如图 1-6 所示。

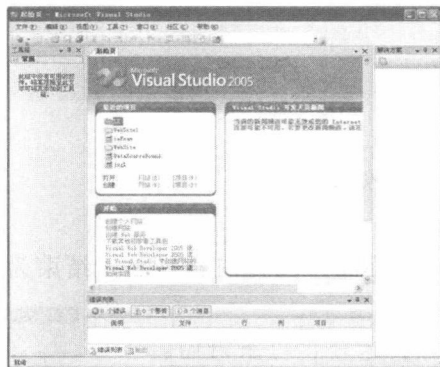


图 1-6 Visual Studio 2005 系统的初始界面

## 1.5 第一个 ASP.NET 程序

网站是管理 Web 应用程序并向外发布信息的基本单位。在 ASP.NET 中，一个站点就是一个 Web 应用程序。每个 ASPX 网页中实际上包含两方面的代码：用于定义显示的代码和用于逻辑处理的代码。用于显示的代码包括 HTML 标记以及对 Web 控件的定义等；用于逻辑处理的代码主要是用 C#. NET（或其他语言）编写的事件处理程序。在 ASPX 网页中，这些代码可以用两种模式存储：一种是代码分离模式；另一种是单一文件模式。

### 1.5.1 代码分离模式

在代码分离模式中，将用于显示的代码（HTML 标记、服务器控件的定义等）放在后缀为 .aspx 的文件中，而用于逻辑处理的代码放到另一个文件中，该文件的后缀依据使用的程序语言而确定。如果使用 C#.NET 语言，则文件的后缀是“aspx.cs”；如果使用 VB.NET 语言，则文件的后缀是“aspx.vb”。此文件有时又可称为代码隐藏（Code-Behind）文件。

【例 1-3】下面程序在用户输入姓名并确定后，将显示欢迎信息，运行结果如图 1-7 和图 1-8 所示。

在文本框中输入“小王”，单击“确定”按钮，结果如图 1-8 所示。



图 1-7 运行初始页面



图 1-8 运行结果

(1) 单击“开始”→“程序”→“Microsoft Visual Studio 2005”→“Microsoft Visual

Studio 2005”，启动 VS 2005。

(2) 单击“文件”→“新建”→“网站”，如图 1-9 所示。

(3) 在“新建网站”对话框中，对各参数进行设置，如图 1-10 所示。

模板：选择“ASP.NET 网站”模板。

“位置”下拉列表：选择网站类型为“文件系统”。

“位置”文本框：选择网站根目录为“C:\WebSite1”。

“语言”下拉列表：选择语言为“Visual C#”。

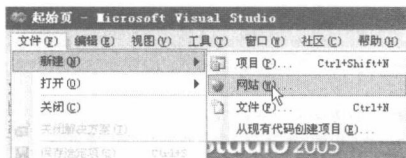


图 1-9 新建网站

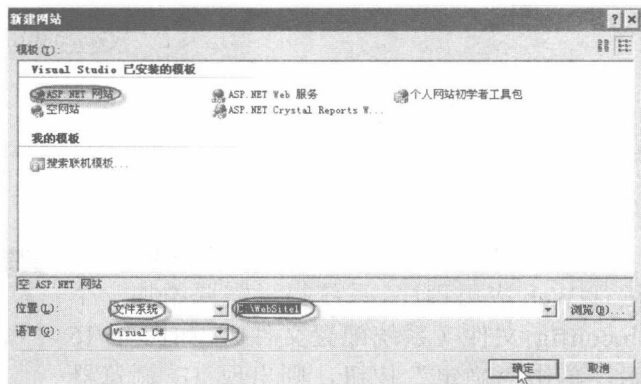


图 1-10 “新建网站”对话框

(4) 单击“确定”按钮，完成文件系统的 ASP.NET 站点建立，进入 Default.aspx 的“源”视图编辑界面，如图 1-11 所示。

(5) 单击界面下方的“设计”选项卡，切换到 Default.aspx 的“设计”视图。从“工具箱”中向设计界面拖放 1 个 TextBox 控件和 1 个 Button 控件；用鼠标选中 Button 控件，按<F4>键，弹出“属性”界面，设置 Button 控件的“Text”属性为“确定”，如图 1-12 所示。



图 1-11 “源”视图编辑界面

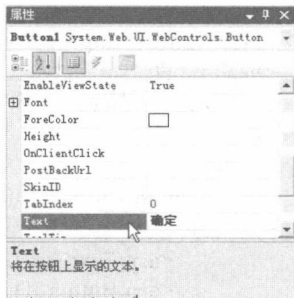


图 1-12 “属性”界面

设计完成的 Default.aspx 界面如图 1-13 所示。

(6) 双击“确定”按钮，切换到代码文件 Default.aspx.cs 的编辑界面，将光标定位于 Button1\_Click 方法体中，输入“Response.Write (\"欢迎你: \"+TextBox1.Text);”，如图 1-14 所示。

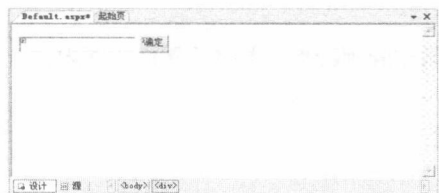


图 1-13 Default.aspx 界面

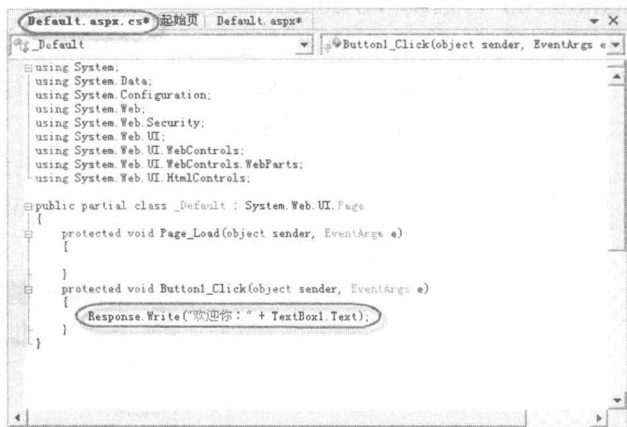


图 1-14 Default.aspx.cs 编辑界面

(7) 单击“调试”→“启动调试”或按<F5>键运行 Web 应用程序，系统提示需要添加一个 Web.config 文件（系统配置文件），如图 1-15 所示，单击“确定”按钮，则出现 IE 浏览器窗口。

经过以上操作，该页面有两个文件构成：网页文件 Default.aspx 与相应的代码文件 Default.aspx.cs。

### Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>无标题页</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="确定" /></div>
</form>
</body>
</html>
```

程序说明：

1) Button1 控件中的 OnClick="Button1\_Click"指定当该控件被单击时，应该执行名为 Button1\_Click 的函数。

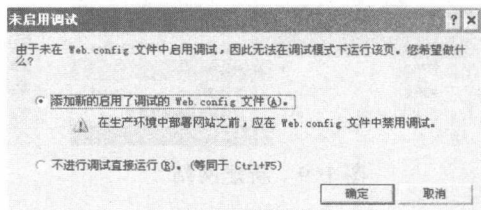


图 1-15 添加 Web.config 文件

2) Web 控件的标记前缀是 `asp:`, 然后指定 Web 控件的类别名称, 如 `asp:Textbox` 和 `asp:Button`。

3) 指定 `runat` 属性为 `server`, 表示控件是在 Server 端执行, 是一个服务器控件。

4) 控件的 `id` 相当于控件的名字; 如果要在代码中引用控件, 就要给控件一个 `id` 属性, 如 `textbox` 控件的 `id` 为 `TextBox1`; 而 `button` 控件在代码中没用到, 可以没有 `id`; 但通常大多数服务器控件都有 `id`。

5) `form` 标记在网页内形成“表单”, 在 ASPX 网页中, 如果 `form` 内有服务器控件, 则 `Form` 标记中一定要加上 `runat=“server”`。

6) 一个页面只能有一个 `form` 表单, 服务器控件应该放在 `form` 表单内。

### Default.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Write("欢迎你: " + TextBox1.Text);
    }
}
```

#### 程序说明:

下面的语句是定义网页类的:

```
public partial class _Default : System.Web.UI.Page
{
    ...
}
```

表明网页是一个类, 派生于 `System.Web.UI.Page` 基类。在类的定义中修饰词“`partial class`”代替了传统的“`class`”, 说明网页是一个“分布式类”。

#### 提示: 什么是分布式类?

有的类比较复杂, 拥有大量的字段、属性、事件和方法, 如果将类的定义都写在一起, 那么文件将非常庞大, 代码的行数很多, 不便于理解和调试。为了降低文件的复杂性, C#. NET 2.0 提供了“分布式类”的概念。在分布式类中, 允许将类的定义分散到多个代码片段之中, 而这些代码片段又可以存放到两个或两个以上的源文件

