高等院校计算机应用系列教材

Web 程序设计
——ASP.NET 网站开发
(第 2 版)

王亚丽　刘金金　主　编
文　坤　程凤娟　副主编

清华大学出版社

北　京

内 容 简 介

本书由浅入深、循序渐进地介绍了使用 ASP.NET 和 Visual Studio 2019 开发环境进行 Web 网站开发所要掌握的各种技术、操作方法和使用技巧。全书共 13 章，分别介绍了 ASP.NET 基础知识、C#入门知识、ASP.NET 服务器控件、验证控制和用户控件、ASP.NET 常用对象、访问数据库、数据绑定、网站设计、LINQ 技术、Web 服务和 ASP.NET AJAX 技术等 Web 网站开发必须了解的各种知识。

本书内容丰富，结构清晰，语言简练，图文并茂，具有很强的实用性和可操作性，是一本适合高等院校 Web 程序设计课程的优秀教材，也可作为广大软件开发人员和系统架构分析人员自学 ASP.NET 的参考书。

图书在版编目(CIP)数据

Web 程序设计：ASP. NET 网站开发 / 王亚丽，刘金金主编. —2 版. —北京：清华大学出版社，2022.1
高等院校计算机应用系列教材
ISBN 978-7-302-59586-1

I. ①W… II. ①王… ②刘… III. ①网页制作工具－程序设计－高等学校－教材 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2021)第 238565 号

责任编辑：王 定
封面设计：高娟妮
版式设计：思创景点
责任校对：成凤进
责任印制：朱雨萌

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
印 装 者：三河市天利华印刷装订有限公司
经 销：全国新华书店
开 本：185mm×260mm **印 张：**20 **字 数：**449 千字
版 次：2012 年 2 月第 1 版 2022 年 2 月第 2 版 **印 次：**2022 年 2 月第 1 次印刷
定 价：59.80 元

产品编号：080152-01

前　言

随着网络技术的不断发展，如何快速而高效地开发出 Web 网站已经成为编程人员所共同关注的问题。为了适应广大编程者对网站开发的需要，Microsoft 公司推出的 ASP.NET 是 Web 应用程序开发平台和框架，实现了企业级 Web 应用程序的快速开发，通过提供简单、具可扩展性的方式，从而开发、部署和运行以任何浏览器或客户端设备作为目标的 Web 应用程序。同时，ASP.NET 支持更多框架和语言的开发，特别是对动态语言的支持，可以让编程人员创建功能更加丰富和界面更加友好的 Web 网站。

本书共分为 13 章，由浅入深、循序渐进地介绍了使用 ASP.NET 开发实用网站所需要掌握的技术，内容结构如下。

第 1 章，ASP.NET 概述。本章介绍了 ASP.NET 框架和网页基础知识、ASP.NET 开发环境 Visual Studio 2019 的安装和使用、ASP.NET 网址的配置，以及 IIS 的安装和配置。通过本章的学习，读者能够对 ASP.NET 有一个初步的认识。

第 2 章，C#入门。本章介绍了开发 ASP.NET 网站所使用的程序设计语言——C#。通过对 C#中关键的语法和面向对象编程知识的学习，读者可为开发 ASP.NET 网站打下良好的基础。

第 3 章，ASP.NET 服务器控件。本章介绍了 ASP.NET 中常用的服务器控件，包括输入控件、输出控件、执行控件和面板控件等常用控件的属性、方法和使用。熟练掌握这些控件的运用方法，读者就可以设计出丰富的网页布局。

第 4 章，验证控件和用户控件。本章内容主要包括验证数据的方法及分类、具体服务验证控件的使用，同时对用户控件的作用和基本开发进行了详细的描述。

第 5 章，ASP.NET 常用对象。本章系统介绍了 ASP.NET 的常用内置对象 Page、Response、Request、Server，以及状态管理对象 Session、Cookie 和 Application。通过使用这些对象的方法和属性，读者可以很方便地实现众多功能。

第 6 章，访问数据库。所有网站的开发离不开数据库，本章重点介绍了 ADO.NET 对 SQL Server 关系型数据库的访问和操作，详细地介绍了整个操作数据库的步骤，使读者能快速掌握访问数据库的方法。

第 7 章，数据绑定。要在网站页面呈现数据内容离不开数据绑定技术，本章首先由浅入深地介绍了数据源绑定技术，然后详细介绍了 SqlDataSource、GridView、ListView、DetailsView 等控件的使用。

第 8 章，网站设计。本章分别介绍了 3 种实现网站页面设计的必备技术，即网站导航、主题和母版页。母版页对整个网站的布局风格和界面的设计统一发挥着重要的作用，主题对于页

面控件的样式控制提供了极佳的帮助，而网站导航则通过对 ASP.NET 中提供的网站地图和常用的 3 种导航控件实现良好的页面导航功能。

第 9 章，LINQ 技术。LINQ 集成查询技术代替了原有的 SQL 语句，可以提供更好的完全面向对象开发的查询。本章将带领读者学习 LINQ 基础知识，掌握 LINQ 的查询语法。读者通过学习 LINQ 中的 LINQ To SQL 技术，可掌握对数据库数据的便捷操作。

第 10 章，Web 服务。本章介绍了 Web 服务的基本原理、各种协议，以及在网站中如何创建、测试和调用引用 Web 服务，其中包括使用存在的 Web 服务和自定义的 Web 服务。

第 11 章，ASP.NET AJAX 技术。本章介绍了风靡一时的 ASP.NET AJAX 技术，内容主要包括 ASP.NET AJAX 的结构组成、核心控件的使用、AJAX Control Toolkit 的 ASP.NET AJAX 扩展控件包，引领读者快速地进入 ASP.NET AJAX 的殿堂。

第 12 章，文件操作。对文件的操作会贯穿整个开发过程，本章将告诉读者如何使用 ASP.NET 4.0 中的各种类来对磁盘、目录、文件、文本等进行操作和读写。

第 13 章，Web 开发应用——办公自动化系统。为更好地加深读者对 ASP.NET 的理解，本章首先从最基本的系统分析与设计开始，确定系统的需求分析和模块划分；然后根据需求分析进行数据库和数据表的结构设计，在此基础上创建出系统的实体类；最后对主要模块界面的设计和实现业务逻辑的代码进行系统的介绍。通过案例的学习，读者能够切实了解一个实际项目开发的流程和所使用的技术。

本书主要有以下特点。

- 理论与实际紧密结合：本书在介绍理论知识的同时，每一章都给出了针对性的案例讲解，力求让读者在掌握基础知识后能够快速上手并举一反三。在每章末尾都配有相应的习题，方便读者课后的实践练习。
- 提供配套的教学资源：为了方便读者自学和教师教学，本书配套有免费电子课件、实例源代码、习题参考答案等，读者可扫二维码获取。


电子课件


实例源代码


习题参考答案

- 内容循序渐进，操作步骤详细：在具体介绍 Visual Studio 功能和操作时，本书提供了每一个功能的具体实现步骤，让读者能够快速了解整个功能的实现方法。案例中的每一个步骤都以通俗易懂的语言进行讲述，读者只需要按照步骤操作，就可以轻松完成知识点的学习。

本书既可以作为高等院校计算机及相关专业学生学习网站开发的技术教材，也可作为广大软件开发人员和系统架构分析设计人员自学 ASP.NET 的参考书。

本书由王亚丽、刘金金任主编，文坤、程凤娟任副主编。由于作者水平有限，书中不足之处在所难免，欢迎广大读者、同仁批评指正。

作　者

2021 年 9 月

目　录

第1章
ASP.NET概述

ASP.NET(Active Server Page.NET)是微软公司推出的基于.NET 框架的新一代网络编程语言，也是目前最新的 Web 技术之一。ASP.NET 开创了公共语言运行库和动态语言运行库相结合的编程框架，可用于在服务器上生成功能强大的 Web 应用程序。本章将介绍 ASP.NET 的相关基础知识以及如何创建其开发环境，使读者对这一强大的 Web 编程工具有一个基本的认识。

☑ 本章重点

- 了解 ASP.NET 的基本框架
- 掌握 IIS 服务器的安装和配置
- 熟悉 Visual Studio 2019 开发环境
- 了解 Web.config 文件的结构

1.1 ASP.NET 框架

.NET 框架是微软公司于 2002 年正式发布的新一代系统、服务和编程平台。它把原有的重点从连接到互联网的单一网站或设备转移到计算机、设备和服务群组上，从而将互联网本身作为新一代操作系统的基础。这样，用户就能够通过控制信息的传递方式、传递时间和传递内容来得到更多的服务。历时 8 年的发展，.NET 技术受到越来越多编程人员的认可。在经历.NET 3.5 的短暂过渡之后，.NET 4.0 正式版本问世了，它的出现代表着一系列可以用来帮助我们建立丰富应用程序的技术又向前发展了一步。

1.1.1 .NET 支持的语言

.NET 框架支持多种语言，包括 C#、VB、J#和 C++等，本书在后台使用的语言主要是 C#。C#是在.NET 1.0 中开始出现的一种新语言，在语法上，它与 Java 和 C++比较相似。实际上 C#是微软整合了 Java 和 C++的优点而开发出来的一种语言，也是微软对抗 Java 平台的一个有效工具。

在被执行之前，所有.NET 语言都会被编译成为一种低级别的语言，这种语言就是中间语言(Intermediate Language，IL)。CLR(Common Language Runtime，公共语言运行时)之所以支持很

多种语言，是因为这些语言在运行之前被编译成了中间语言。因为所有的.NET 语言都建立在中间语言之上，所以 VB 和 C#具有相同的特性和行为。因此，一个使用 C#编写的 Web 页面也可以使用 VB 编写的组件，同样使用 VB 编写的 Web 页面也可以使用 C#编写的组件。

.NET 框架提供了一个公共语言规范(Common Language Specification，CLS)以保证这些语言之间的兼容性。只要遵循 CLS，任何利用某一种.NET 语言编写的组件都可以被其他语言所引用。CLS 的一个重要部分是公共类型系统(Common Type System，CTS)，CTS 定义了诸如数字、字符串和数组等数据类型的规则，这样它们就能为所有的.NET 语言所共享。CLS 还定义了诸如类、方法、实践等对象成分。事实上，基于.NET 进行程序开发的程序员不需要考虑 CLS 是如何工作的，因为这一切都由.NET 4.0 平台自动完成。CLR 只执行中间语言代码，然后把它们进一步编译成为机器语言代码，以能够使当前平台所执行。

1.1.2 公共语言运行时

公共语言运行时是指用.NET 语言编写的代码公共运行环境。它既是.NET 框架的基础，也是实现.NET 跨平台、跨语言、代码安全等核心特性的关键。公共语言运行时就像一个执行程序时管理代码的代理，以跨语言集成、自描述组件、简单配制和版本化及集成安全服务为特点，提供核心服务(如内存管理、线程管理和远程处理)。

公共语言运行时管理的.NET 中的代码，称为受托管代码。它们包含了有关代码的信息，例如代码中定义的类、方法和变量。受托管代码中所包含的信息称为元数据。公共语言运行时使用元数据来安全地执行代码程序。除了安全地执行程序以外，受托管代码的目的在于 CLR 服务。这些服务包括查找和加载类，以及与现有的动态链接库(Dynamic Link Library，DLL)代码和组件对象之间的相互操作。

公共语言运行时遵循公共语言架构的标准，可以使 C++、C#、Visual Basic 及 JScript 等多种语言深度集成。

1.1.3 动态语言运行时

从.NET 4 版本起，框架中增加了动态语言运行时(Dynamic Language Runtime，DLR)的新特性。就像公共语言运行时为静态型语言(如 C#和 Visual Basic)提供了通用平台一样，动态语言运行时为动态型语言(如 JavaScript、Ruby、Python)甚至 COM 组件等提供了通用平台，这代表.NET 4 框架在互操作性方面向前迈进了一大步。

动态语言运行时是一种运行时环境，它将一组适用于动态语言的服务添加到公共语言运行时。借助于动态语言运行时，开发人员可以更轻松地开发要在.NET 框架上运行的动态语言，而且向静态类型化语言添加动态功能也会变得更容易。

动态语言运行时的目的是允许动态语言系统在.NET 框架上运行，并为动态语言提供.NET 互操作性。在 Visual Studio 中，动态语言运行时将动态对象引入到 C#和 Visual Basic 中，以便这些语言能够支持动态行为，并且可以与动态语言进行互操作，同时动态语言运行时还可帮助用户创建支持动态操作的库。

与公共语言运行时类似，动态语言运行时是.NET 4 框架的一部分，并随.NET Framework 和 Visual Studio 安装包一起提供。

动态语言运行时通过在调用站点中使用联编程序，不仅可以与.NET Framework 通信，还可以与其他基础结构和服务(包括 Silverlight 和 COM)通信。联编程序将封装语言的语义，并指定如何使用表达式在调用站点中执行操作。这样，使用动态语言运行时的动态和静态类型化语言就能够共享类库，并获得对动态语言运行时支持的所有技术的访问权。

1.1.4　.NET 类库

.NET 框架的另一个主要组件是类库，它是一个综合性的面向对象的可重用类型集合，例如 ADO.NET、ASP.NET 等。.NET 基类库位于公共语言运行库的上层，与.NET Framework 紧密集成在一起，可被.NET 支持的任何语言所使用。这也是 ASP.NET 中可以使用 C#、VB.NET、VC.NET 等语言进行开发的原因。.NET 类库非常丰富，提供数据库访问、XML、网络通信、线程、图形图像、加密等多种功能服务。类库中的基类提供了标准的功能，如输入输出、字符串操作、安全管理、网络通信、线程管理、文本管理和用户界面设计功能。这些类库使得开发人员更容易建立应用程序和网络服务，从而提高开发效率。

1.2　网页基础知识

要开发一个网站，首先要了解组成网站的最基本的元素——网页。本节就来了解一下网页的基础知识，包括网页和服务器的交互过程、静态页面和动态网页以及脚本语言。

1.2.1　网页和服务器的交互

通过互联网浏览网页时，用户会自动与网页服务器建立连接。用户提交信息资源的过程称为向服务器发出请求。通过服务器解释信息资源来定位对应的页面，并传送回代码来创建页面，这个过程称为对浏览器的响应。浏览器接收来自网页服务器的代码，并将它编译成可视页面。在这样的交互过程中，浏览器称为“客户机”或者“客户端”，整个交互的过程则称为“客户机/服务器”的通信过程。

“客户机/服务器”通信过程概括了任务的分布来描述网页的工作方式。服务器(Web 服务器)存储、解释和分布数据，客户机(浏览器)访问服务器以得到这些数据。客户机和服务器使用 HTTP 协议通过 Internet 进行交互。HTTP 协议又叫作超文本传输协议，是一个客户机和服务器端请求和应答的标准。浏览网页时，浏览器通过 HTTP 协议与服务器交换信息。

1.2.2　静态页面

早期的网站发布的是静态网页，主要由 HTML 语言组成，没有其他可以执行的程序代码。静态页面一经制成，内容就不会再改变，不管何时何人访问，显示的都是一样的内容，如果要修改有关内容，就必须修改源代码，然后重新上传到服务器上。静态页面虽然包含文字和图片，但这些内容需要在服务器端以手工的方式来变换，因此很难把它们描述为 Web 程序。下面是使用 HTML 语言编写的一个简单静态网页代码。

代码说明：该程序包含一个标题和一行文字。其中，标题包含在标记<h1>和</h1>之间，文字包含在标记 和 之间。图 1-1 显示了该静态网页文件被浏览器解析后的结果。

```
<html>
<head>
    <title>这是一个静态网页</title>
</head>
<body>
    <h1>这是一个静态网页</h1>
    <p>静态网页使用 HTML 语言编写</p>
</body>
</html>
```

图 1-1　静态网页示例

HTML 是互联网的描述语言，基本的 HTML 语言包含由 HTML 标记格式化的文本和图像内容。文本是 HTML 要显示的内容，标记则告诉浏览器如何显示这些内容，它定义了不同层次的标题、段落、链接、格式等。HTML 文件的后缀可以是.htm，也可以是.html。

1.2.3　动态页面

动态页面不仅含有 HTML 标记，而且含有可以执行的程序代码。动态页面能够根据不同的输入和请求动态生成返回的页面，例如常见的 BBS、留言板、聊天室等就是用动态网页来实现的。动态网页的使用非常灵活，功能强大。

代码说明：该程序由 HTML 表单组成，包括一个标题、四个复选框和一个提交按钮，这些内容和标记均被包含在表单标记之间。该网页运行效果如图 1-2 所示。

```
<html>
  <head>
    <title>这是一个动态网页</title>
</head>
 <body>
   <form>
        <h3>请选择您喜欢的水果？</h3>
        <p>请做出选择：</p>
        <input type="checkbox" />苹果<br/>
        <input type="checkbox" />香蕉<br/>
        <input type="checkbox" />芒果<br/>
        <input type="checkbox" />提子<br/>
        <input type="submit" value="提  交">
   </form>
 </body>
</html>
```

图 1-2　动态网页示例

网页是构成网站的基本元素，是承载各种网站应用的平台。网页的文件格式通常为 HTML 格式，可以以文件的形式存储在世界某个角落的某一台计算机中。

1.2.4　脚本语言

在网页的发展过程中出现了很多优秀的脚本语言，如 ASP、JSP、PHP 等。脚本语言确实简化了 Web 程序的开发，但其使用起来也有很大的缺点。首先，它的代码和 HTML 标记杂乱地堆砌在一起，显得很混乱，非常不方便开发和维护，所以当 ASP.NET 的代码和 HTML 标记分离后，使用以往一些脚本语言的 Web 开发人员都有一种耳目一新的感觉；其次，脚本语言的编程思想不符合当前流行的面向对象编程思想。基于以上原因，脚本语言必将会被其他更高级语言(如 ASP.NET 和 Java 等)所代替。

1.3　ASP.NET 应用程序

ASP.NET 应用程序是一系列资源和配置的组合，这些资源和配置只在同一个应用程序内共享，而其他应用程序则不能享用这些资源和配置，即使它们发布在同一台服务器上。就技术而言，每个 ASP.NET 应用程序都运行在一个单独的应用程序域。应用程序域是内存中的独立区域，这样可以确保在同一台服务器上的应用程序不会相互干扰，也可以避免因为其中一个应用程序发生错误就影响到其他应用程序的正常进行。同样，应用程序域限制一个应用程序中的 Web 页面访问其他的应用程序的存储信息。每个应用程序单独运行，具有自己的存储、应用和会话数据。

ASP.NET 应用程序的标准定义是：文件、页面、处理器、模块和可执行代码的组合，并且它们能够从服务器上一个虚拟目录中被引用。换句话说，虚拟目录是界定应用程序的基本组织结构。

1.3.1　ASP.NET 页面与服务器交互

ASP.NET 页面作为代码在服务器上运行。在用户单击按钮(或者当用户选中复选框或与页面中的其他控件交互)时，页面被提交到服务器。每次页面都会回发，以便它可以再次运行其服务器代码，然后向用户呈现其自身的新版本。传递 Web 页面的具体过程如下。

(1) 用户通过客户端浏览器请求页面。使用 HTTP GET 方法请求页面，页面第一次运行，执行初步处理。

(2) 页面将标记动态呈现到浏览器。

(3) 用户输入信息或从可用选项中进行选择，然后单击按钮。如果用户单击链接而不是按钮，页面可能仅仅定位到另一页，而第一页不会被进一步处理。

(4) 页面发送到 Web 服务器。浏览器执行 HTTP POST 方法，该方法在 ASP.NET 中称为“回发”。更明确地说，页面发送回其自身。例如，如果用户正在使用 Default.aspx 页面，则单击该页上的某个按钮可以将该页发送回服务器，发送的目标则是 Default.aspx。

(5) 在 Web 服务器上，该页再次运行，并且可在页面上使用用户输入或选择的信息。

(6) 页面执行通过编程所要实行的操作，服务器将执行操作后的页面以 HTML 标记的形式发送到客户端浏览器。

只要用户在该页面中工作，此循环就会继续。用户每次单击按钮时，页面中的信息会发送到 Web 服务器，然后该页面再次运行。每个循环称为一次“往返行程”。由于页面处理发生在

Web 服务器上，因此页面可以执行的每个操作都需要一次到服务器的往返行程。

1.3.2 ASP.NET Web 窗体

在 ASP.NET 中，发送到客户端浏览器中的网页是经过.NET 框架中的基类动态生成的。这个基类就是 Web 页面框架中的 Page 类，而实例化的 Page 类就是一个 Web 窗体，也就是 Web Forms。因此，一个 ASP.NET 页面就是一个 Web 窗体。窗体对象也都具有其属性、方法和事件，可以作为容器容纳其他控件。

Web 窗体是一个后缀名为.aspx 的文本文件，可以使用任何文本编辑器打开和编写它。ASP.NET 是编译的运行机制，为了简化开发人员的工作，一个.aspx 页面不需要手工编译，而是在页面被调用时，由公共语言运行时自行决定是否要被编译。

Web 窗体可以使用一般的 HTML 窗体控件，但 ASP.NET 也提供了可以在服务器上运行的 Web 窗体控件。

1.3.3 后台隐藏代码页

早期脚本语言是将代码和 HTML 标记混合在一起编写，而后台隐藏代码页则与其不同。它是将业务逻辑的处理代码都存放在.cs 文件中，当 ASP.NET 网页运行时，ASP.NET 类生成时会先处理.cs 文件中的代码，再处理.aspx 页面中的代码，这种过程称为代码分离。

代码分离的优点是在.aspx 页面中，开发人员可以将页面直接作为样式来设计，即美工人员可以设计.aspx 页面，而.cs 文件由编程人员来完成业务逻辑的处理。同时，将 ASP.NET 中的页面样式代码和逻辑处理代码分离，能够让维护变得简单并且代码看上去整洁明了。

1.3.4 ASP.NET 新特性

与之前的技术相比较而言，在新版本的.NET 架构中，Microsoft 使得 ASP.NET 家族有了崭新的面貌。除了保留.NET Framework、ASP.NET 技术而外，Microsoft 引入了.NET Core 和应用于 Web 开发的 ASP.NET Core，下面主要对 ASP.NET Core 进行简要介绍。

1. ASP.NET Core 概述

ASP.NET Core 是一个跨平台的高性能开源框架，用于生成基于云且连接 Internet 的新式应用程序。使用 ASP.NET Core 可以完成以下工作。

- 创建 Web 应用和服务、IoT 应用和移动后端。
- 在 Windows、macOS 和 Linux 上使用喜爱的开发工具。
- 部署到云或本地。
- 在.NET Core 或.NET Framework 上运行。

2. ASP.NET Core 的优点

许多开发人员使用 ASP.NET 4.x 创建 Web 应用。而 ASP.NET Core 是对 ASP.NET 4.x 的重新设计，通过体系结构上的更改，产生了更精简、更模块化的框架。ASP.NET Core 具有如下优点。

(1) 生成 Web UI 和 Web API 的统一场景。

(2) 针对可测试性进行构建。

(3) Razor Pages 可以使基于页面的编码方式更简单高效。

(4) Blazor 允许开发人员在浏览器中使用 C#和 JavaScript，共享使用.NET 编写的服务器端和客户端应用逻辑。

(5) 能够在 Windows、macOS 和 Linux 上进行开发和运行。

(6) 开放源代码和以社区为中心。

(7) 集成新式客户端框架和开发工作流。

(8) 支持使用 gRPC 托管远程过程调用(RPC)。

(9) 提供基于环境的云就绪配置系统。

(10) 内置依赖项注入。

(11) 提供轻型的高性能模块化 HTTP 请求管道。

(12) 能够托管于以下各项：Kestrel、IIS、HTTP.sys、Nginx、Apache、Docker。

(13) 能够并行版本控制。

(14) 提供简化新式 Web 开发的工具。

3. ASP.NET Core MVC

使用 ASP.NET Core MVC 生成 Web API 和 Web UI。ASP.NET Core MVC 提供生成 Web API 和 Web 应用所需的功能。

(1) Model-View-Controller (MVC)模式使 Web API 和 Web 应用可测试。

(2) Razor Pages 是基于页面的编程模型，它让 Web UI 的生成更加简单高效。

(3) Razor 标记提供了适用于 Razor 页面和 MVC 视图的高效语法。

(4) 标记帮助程序使服务器端代码可以在 Razor 文件中参与创建和呈现 HTML 元素。

(5) 内置的多数据格式和内容协商支持使 Web API 可访问多种客户端，包括浏览器和移动设备。

(6) 模型绑定自动将 HTTP 请求中的数据映射到操作方法参数。

(7) 模型验证自动执行客户端和服务器端验证。

4. 面向.NET Framework 的 ASP.NET Core

ASP.NET Core 2.x 可以面向.NET Core 或.NET Framework。面向.NET Framework 的 ASP.NET Core 应用无法跨平台，它们仅在 Windows 上运行。通常，ASP.NET Core 2.x 由.NET Standard 库组成。使用.NET Standard 2.0 编写的库在实现.NET Standard 2.0 的任何.NET 平台上运行。

ASP.NET Core 2.x 在以下实现.NET Standard 2.0 的.NET Framework 版本上受支持：

- 强烈建议使用最新版本的.NET Framework。
- .NET Framework 4.6.1 及更高版本。

ASP.NET Core 3.0 以及更高版本只能在.NET Core 中运行，有关详细信息请参阅 *A first look at changes coming in ASP.NET Core 3.0*(《抢先了解 ASP.NET Core 3.0 即将推出的更改》)。

与.NET Framework 相比，.NET Core 有以下几个优势，并且这些优势会随着每次发布增加。

- 跨平台，可在 macOS、Linux 和 Windows 上运行。
- 性能更强。

- 并行版本控制。
- 新 API。
- 开源。

Microsoft 也正在努力缩小 API 在.NET Framework 与.NET Core 中的差距。Windows 兼容性包使数千个仅可在 Windows 运行的 API 可在.NET Core 中使用。这些 API 在.NET Core 1.x 中不可用。

5. 在 ASP.NET 4.x 和 ASP.NET Core 之间进行选择

与 ASP.NET Core 相比，ASP.NET 4.x 是一个成熟的框架，提供在 Windows 上生成基于服务器的企业级 Web 应用所需的服务。

表 1-1 将 ASP.NET Core 与 ASP.NET 4.x 进行了详细的比较。

表 1-1　ASP.NET Core 与 ASP.NET 4.x 应用特点对照表

ASP.NET Core	ASP.NET 4.x
针对 Windows、macOS 或 Linux 进行生成	针对 Windows 行生成
Razor 页面是在 ASP.NET Core 2.x 及更高版本中创建 Web UI 时建议使用的方法	使用 Web Forms、SignalR、MVC、Web API、WebHook 或网页
每个计算机多个版本	每个计算机有一个版本
使用 C#或 F#通过 Visual Studio、Visual Studio for Mac 或 Visual Studio Code 进行开发	使用 C#、VB 或 F#通过 Visual Studio 进行开发
比 ASP.NET 4.x 性能更高	良好的性能
使用.NET Core 运行时	使用.NET Framework 运行时

6. .NET Core 和.NET Framework 在服务器端应用程序开发上的差异

对服务器应用程序使用.NET Core 的情况如下：

- 跨平台需求。
- 面向微服务。
- 使用 Docker 容器。
- 高性能和可扩展的系统。
- 需按应用程序提供并行的.NET 版本。

对服务器应用程序使用.NET Framework 的情况如下：

- 应用当前使用.NET Framework(建议扩展而不是迁移)。
- 应用使用不可用于.NET Core 的第三方.NET 库或 NuGet 包。
- 应用使用不可用于.NET Core 的.NET 技术。
- 应用使用不支持.NET Core 的平台。Windows、macOS 和 Linux 支持.NET Core。

7. ASP.NET 2019 的改进

Visual Studio 2019 开发环境中的网页设计器已进行改进，提高了 CSS 的兼容性，增加了对 HTML 和 ASP.NET 标记代码段的支持，并提供了重新设计的 JScript 智能感知功能。

1.4 建立 ASP.NET 开发和运行环境

要使用.NET 4.0 框架来开发网站程序，需要先建立 ASP.NET 开发和运行的环境，这一过程包括安装和配置 IIS Web 服务器以及安装 Visual Studio 2019 开发环境。

1.4.1 安装和配置 IIS Web 服务器

IIS 是 Internet Information Server 的缩写，是微软公司主推的 Web 服务器。通过 IIS，开发人员可以方便地调试程序或发布网站。

在 Windows 10 中安装和配置 IIS 服务器的具体步骤如下。

(1) 在 Windows 系统中打开如图 1-3 所示的“程序和功能”中的“卸载或更改程序”窗口，该窗口显示当前已经安装的程序。

(2) 在窗口的左侧选择“启用或关闭 Windows 功能”图标，弹出如图 1-4 所示的“Windows 功能”对话框，找到 Internet Information Services 复选框，如果尚未安装，则其左侧的复选框不会被选中；如果复选框是不可选状态，说明 IIS 的组件没有全部安装。

图 1-3　“卸载或更改程序”窗口

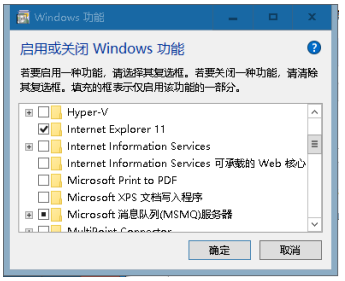

图 1-4　“Windows 功能”对话框

(3) 如果复选框 Internet Information Services 没有被选中，则选中该复选框。也可进一步单击左边“+”号展开可选项，如图 1-5 所示。

(4) 选择要安装的选项后，单击“确定”按钮，完成 IIS 的安装。

(5) 选择“开始”|“控制面板”|“管理工具”|“IIS 管理器”命令，弹出如图 1-6 所示的“Internet Information Services(IIS)管理器”窗口，依次展开左侧的“根”节点(应用程序池)、“网站”节点(网站)、“默认网站”节点(Default Web Site)。

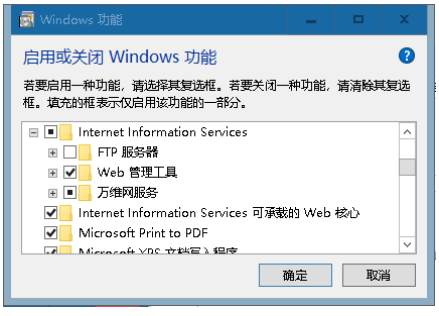

图 1-5　展开可选项

图 1-6　“Internet Information Services(IIS)管理器”窗口

(6) 右击“默认网站”节点 Default Web Site，弹出如图 1-7 所示的菜单。用户可以选择“管理网站”下的“停止”命令关闭 IIS 服务，也可以选择“重新启动”命令重启 IIS 服务。

说明：当用户通过 HTTP 浏览位于 Web 服务器上的一些 Web 页面时，Web 服务器需要确定与该页面对应的文件位于服务器硬盘上的位置。事实上，在由 URL 给出的信息与包含页面文件的物理位置(在 Web 服务器的文件系统中)之间有着重要的关系。这个关系是通过虚拟目录来实现。

虚拟目录相当于物理目录在 Web 服务器机器上的别名，它不仅使用户避免了冗长的 URL，也是一种很好的安全措施，因为虚拟目录对所有浏览者隐藏了物理目录结构。

(7) 在硬盘上创建一个物理目录，这里在 D 盘的根目录下创建一个目录，命名为 WebTest。

(8) 启动“Internet Information Services(IIS)管理器”窗口，右击“默认网站”节点 Default Web Site，选择“添加虚拟目录”命令，弹出如图 1-8 所示的“添加虚拟目录”对话框。

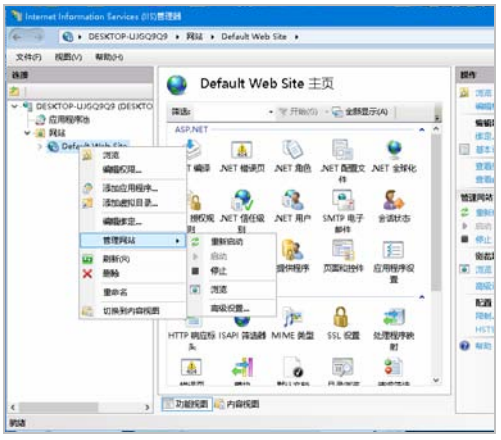

图 1-7　选择菜单

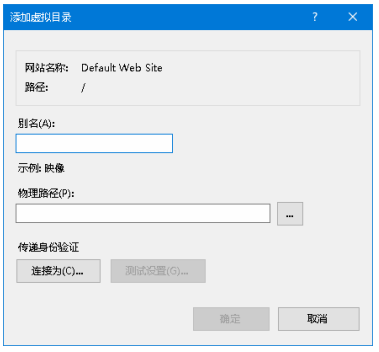

图 1-8　“添加虚拟目录”对话框

(9) 如图 1-9 所示填写虚拟目录别名，在“别名”文本框中输入虚拟目录的名字 WebTest，和它的物理目录的名字相同。选择刚才创建的物理目录 D:\WebTest，如图 1-10 所示。

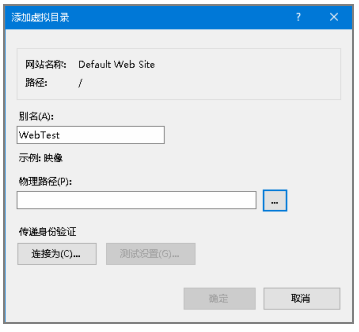

图 1-9　填写虚拟目录别名

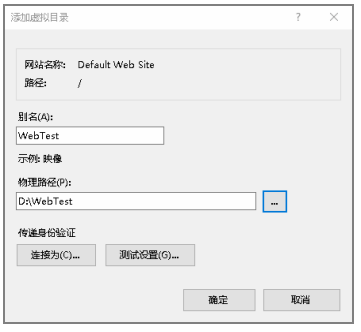

图 1-10　选择物理路径

(10) 单击“确定”按钮，完成虚拟目录的创建。此时，在“Internet Information Services(IIS)管理器”窗口的目录中将显示该 WebTest 虚拟目录，如图 1-11 所示。

(11) 选择“功能视图”选项卡，并双击“默认文档”图标，如图 1-12 所示，打开如图 1-13 所示的“默认文档”界面，用户可以在打开的选项区域管理默认文档。

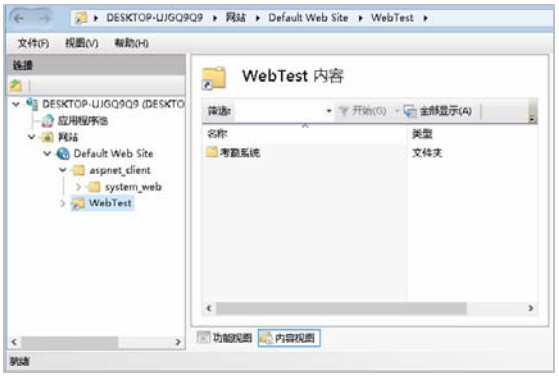

图 1-11　显示虚拟目录

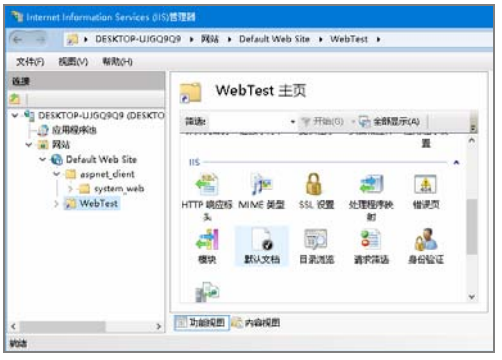

图 1-12　“默认文档”设置

(12) 在图 1-12 所示的对话框中单击“身份验证”图标，打开如图 1-14 所示的“身份认证”界面，用户可以在打开的选项区域设置身份验证属性。

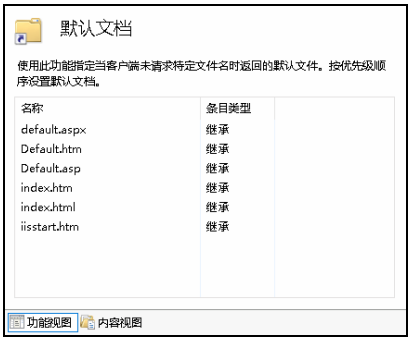

图 1-13　管理默认文档

图 1-14　设置身份验证属性

1.4.2　Visual Studio 2019 开发环境

每一个正式版本的.NET 框架都有一个与之对应的高度集成的开发环境，微软称之为 Visual Studio，中文意思是可视化工作室。随同 ASP.NET 一起发布的开发工具是 Visual Studio 2019。它对基于 ASP.NET 的项目开发有很大帮助，使用 Visual Studio 2019 可以很方便地进行各种项目的创建、具体程序的设计、程序调试和跟踪以及项目发布等。

1. 安装 Visual Studio 2019 开发环境

Visual Studio 2019 目前有多个版本：Visual Studio Community 2019、Visual Studio Professional 2019 以及 Visual Studio Enterprise 2019。其中，第一种为免费供学生、开放源代码参与者和个人使用的社区版；第二种用于个人和小型开发团队，采用最新技术开发应用程序和实现有效的商务目标；第三种为需完成体系结构设计、应用开发、数据库开发以及应用程序测试等多任务的团队提供集成的工具集，在应用程序生命周期的每个步骤，团队成员都可以继续协作并利用一个完整的工具集与指南。

本书所用版本为 Visual Studio Community 2019，下面介绍 Visual Studio 2019 Community 的安装过程。

(1) 用户可以访问网址:

https://visualstudio.microsoft.com/zh-hans/thank-you-downloading-visual-studio /?sku=Community&rel=16

下载 Visual Studio 2019 社区版，也可以按照需要购买其他正版的安装程序。

(2) 打开安装程序后，首先进入“安装向导”界面。选择“安装 Microsoft Visual Studio 2019”，即可进入资源复制过程提示框。

(3) 单击“继续”按钮，经过一段时间的等待之后，进入如图 1-15 所示的安装路径的设定界面。

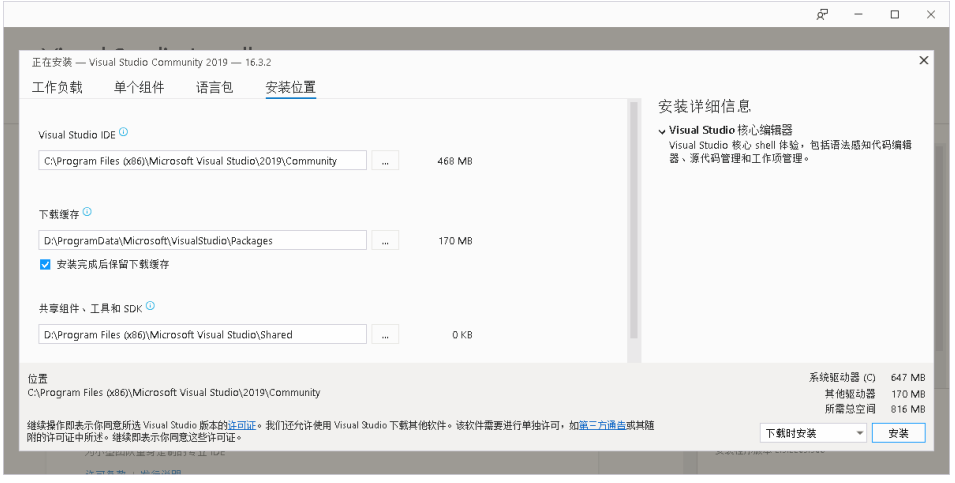

图 1-15　安装路径的设定界面

(4) 在图 1-15 所示界面中单击“安装”之后进入图 1-16 所示的安装功能显示界面，勾选需要的选项。其中 Python 开发与本书内容无关，可不选。

图 1-16　安装功能显示界面

(5) 单击“安装”按钮，开始安装并显示当前安装的组件。当所有组件安装成功后，进入

图 1-17 所示的界面，显示已经成功地安装 Visual Studio 2019，可以单击“启动”按钮，结束安装过程并打开 Visual Studio 2019。

图 1-17　安装完成界面

至此，Visual Studio 2019 成功地被安装到本地计算机上。

2. 创建 Web 项目

安装完成 Visual Studio 2019 开发环境之后，就要使用这一强大的工具来创建一个 ASP.NET 项目，让大家对 Visual Studio 2019 有一个初步的了解。

选择“开始”|“所有程序”| Microsoft Visual Studio 2019 | Microsoft Visual Studio 2019 命令，打开 Visual Studio 2019，进入如图 1-18 所示主界面。

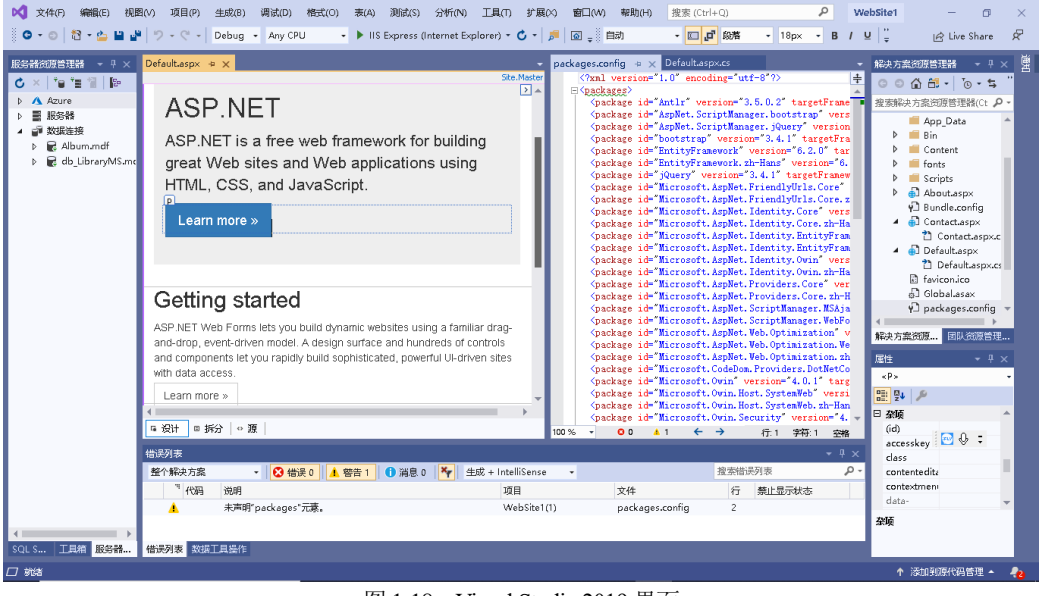

图 1-18　Visual Studio 2019 界面

Visual Studio 2019 主界面中主要组成部分如下。

(1) 菜单栏：位于主界面的顶部，包含了实现软件所有功能的选项。

(2) 工具栏：位于菜单栏的下方，包含了软件常用功能的快捷按钮。

(3) 状态栏：位于主界面的底部，用于显示软件的状态信息。

(4) 工作区：位于主界面中工具栏和状态栏之间的显示部分，占据了主界面的绝大部分位置。显示的内容包括当前项目的界面设计、代码编写、调试等窗口。

(5) 工具箱：位于主界面的左侧栏，提供了设计页面时常用的各种控件，只要简单地将控件拖动到设计页面即可方便地使用。

(6) 解决方案资源管理器：位于主界面的右侧边最上部，用于对解决方案和项目进行统一的管理，其主要组成是各种类型的文件目录。

(7) 团队资源管理器：位于解决方案资源管理器的下方，是一个简化的 Visual Studio Team System 2019 环境，专用于访问 Team Foundation Server 服务。

(8) 服务器资源管理器：位于主界面左侧栏，用于打开数据连接，登录服务器，浏览数据库和系统服务。

单击菜单上的“文件”|“新建项目”命令，打开如图 1-19 所示的“创建新项目”对话框，选择“ASP.NET Web 应用程序”，然后单击“下一步”按钮，弹出“配置新项目”对话框，如图 1-20 所示，在“项目名称”文本框中输入项目名称，并在“位置”文本框中输入相应的存储路径，在“解决方案名称”文本框中输入解决方案名称。最后单击“创建”按钮，即可创建一个新的 Web 项目。

图 1-19 “创建新项目”对话框

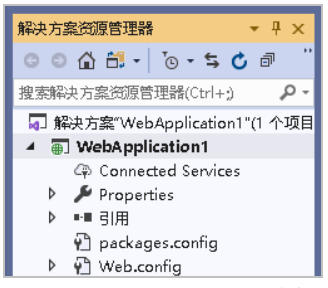

图 1-20 “创建新项目”对话框

3. Web 项目管理

当创建一个新的网站项目之后，用户可以利用解决方案资源管理器对网站项目进行管理。通过解决方案资源管理器，用户可以浏览当前项目所包含的所有资源(.aspx 文件、.cs 文件、图片等)，也可以向项目中添加新的资源，并且可以修改、复制和删除已经存在的资源。解决方案资源管理器如图 1-21 所示。下面主要介绍如何向项目中添加新资源。

在“解决方案资源管理器”中右击项目名称 WebApplication1，会弹出如图 1-22 所示的菜单。

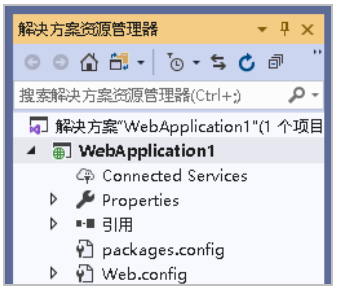

图 1-21　解决方案资源管理器

图 1-22　右击弹出的快捷菜单

如图 1-23 所示，“添加”命令下有多个添加项，分别是“添加”“添加引用”“添加 Web 引用”“添加服务引用”。其中，“添加引用”命令用来添加对类的引用，“添加 Web 引用”命令用来添加对存在于 Web 上的公开类的引用，“添加服务引用”命令用来添加对服务的引用。

选择“添加”命令，弹出下一级子菜单，包括 “新建项”“现有项”“新建文件夹”和“添加 ASP.NET 文件夹”4 个命令。其中，“新建项”命令用来添加 ASP.NET 支持的所有文件资源，“现有项”命令用来把已经存在的文件资源添加到当前项目中去，“新建文件夹”命令用来向网站项目中添加一个文件夹，“添加 ASP.NET 文件夹”命令用来向网站项目中添加一个 ASP.NET 独有的文件夹。

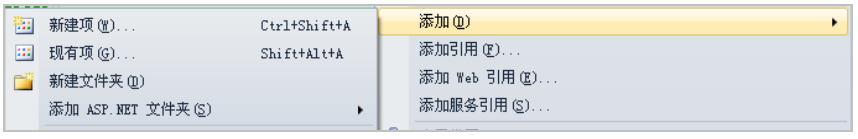

图 1-23　“添加”的子菜单

选择“新建项”命令，打开如图 1-24 所示的“添加新项”对话框，对话框左侧显示“已安装”模板的树状列表，中间显示与选定模板相对应的模板文件列表，右侧是对模板的描述。选择“已安装”模板下的 Web 模板，并在模板文件列表中选中“Web 窗体”，然后在“名称”文本框中输入该文件的名称，最后单击“添加”按钮即可向网站项目中添加一个新的文件。

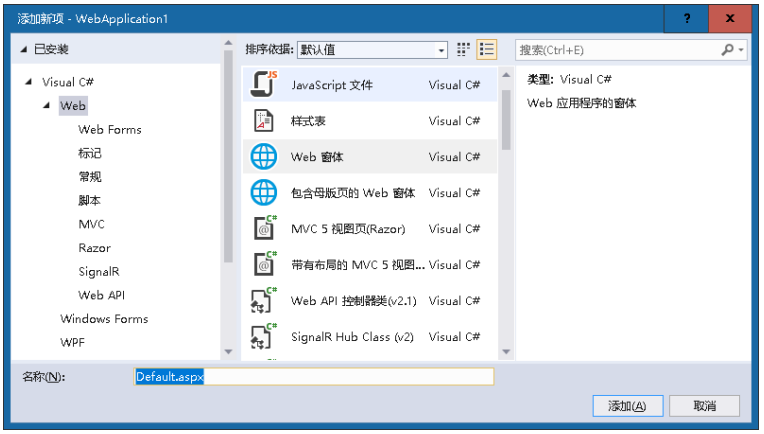

图 1-24　“添加新项”对话框

4. 编辑 Web 页面

在添加一个 Web 页面后，用户可以使用 Visual Studio 对它进行编辑，在解决方案资源管理器中双击某个要编辑的 Web 页面文件，该页面文件就会在视图设计器窗口中打开，如图 1-25 所示。

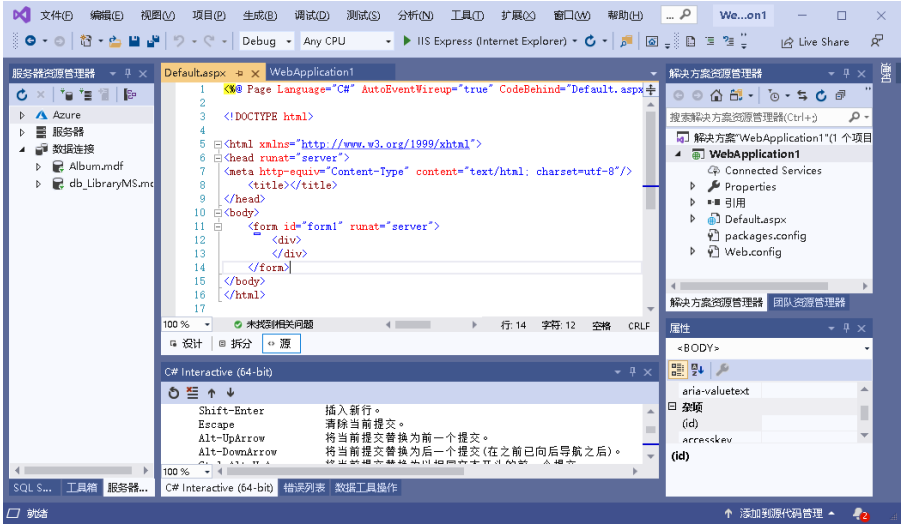

图 1-25　视图设计器

用户可以通过视图设计器窗口底部的“设计”“拆分”和“源”3 个按钮来进行 3 种视图的编辑。其中，设计视图用来显示设计的效果，并且可以从“工具箱”中直接把控件放置在设计视图中，“工具箱”是放置控件的容器，如图 1-26 所示；拆分视图同时显示设计视图和源视图；源视图显示设计源码，开发者可以在该视图中直接通过编写代码来设计页面。

5. 属性查看器

在 Web 页面的设计视图下，右击某一个控件或页面的任何地方，在弹出的菜单中选择“属性”命令或者在菜单栏中选择“视图”|“属性窗口”命令，会弹出如图 1-27 所示的控件“属性”窗口。在“属性”窗口中，可以编辑控件的属性。

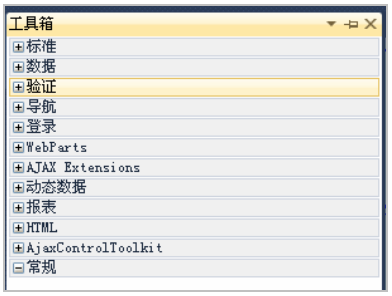

图 1-26　工具箱

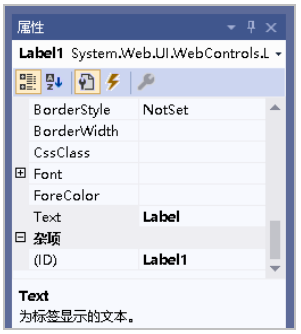

图 1-27　“属性”窗口

6. 编辑后台代码

在 Web 页面的设计视图下，双击页面的任何地方即可打开隐藏的后台代码文件，在此界面中，开发者可以编写与页面对应的后台逻辑代码。或者通过双击网站目录下的文件名，也可以进入后台代码文件，如图 1-28 所示。

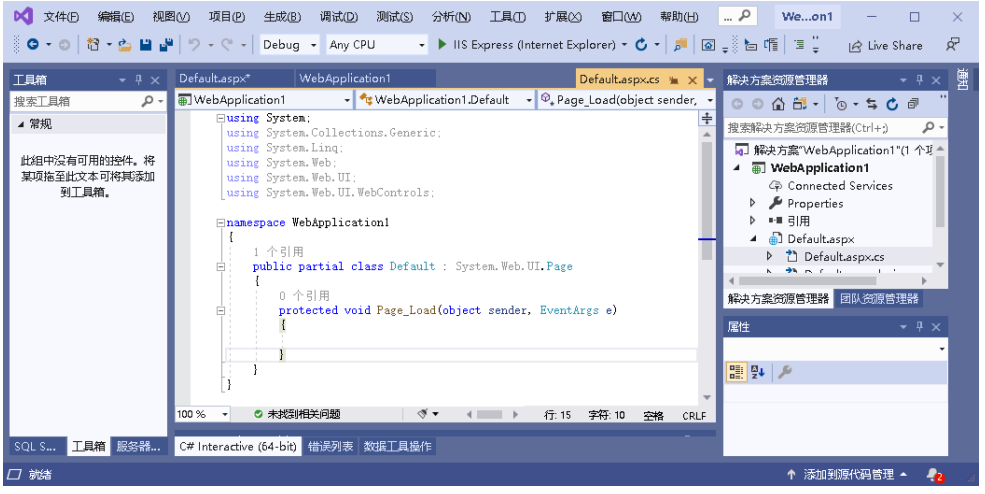

图 1-28　显示后台隐藏的代码文件

7. 编译和运行应用程序

选择“生成”|“生成网站”命令，如果生成成功，则屏幕下方的“输出”窗口中将显示相关信息，如图 1-29 所示。

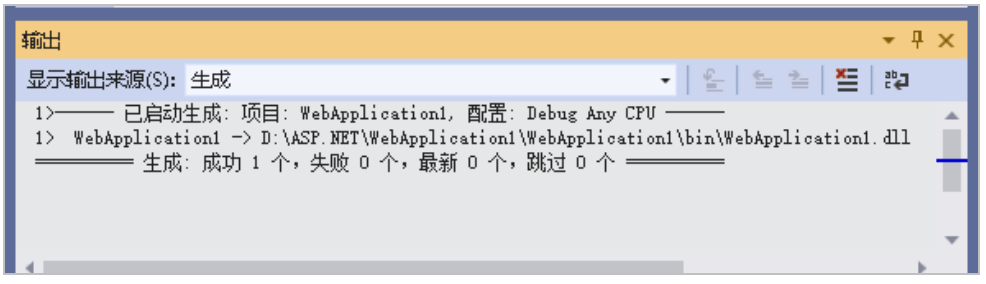

图 1-29　输出窗体

单击工具栏上的“启动调试”按钮运行程序，浏览器就会显示程序的运行效果。

1.4.3 Visual Studio 重要特性

Visual Studio 集成开发环境的重要特性有以下几种。

1. 窗口移动

文档窗口不再受限于集成开发环境(IDE)的编辑框架。开发人员可以将文档窗口停靠在 IDE 的边缘，或者将它们移动到桌面(包括辅助监视器)上的任意位置。如果打开并显示两个相关的文档窗口，则在一个窗口中所做的更改将立即反映在另一个窗口中。

工具窗口也可以进行自由移动，使它们停靠在 IDE 的边缘、浮动在 IDE 的外部或者填充部分或全部文档框架。这些窗口始终保持可停靠的状态。

2. 调用层次结构

调用层次结构可以帮助开发人员分析代码，并实现导航定位功能。在方法、属性、字段、索引器或者构造函数上右击，在弹出菜单中选择“查看调用层次结构”命令。在如图 1-30 所示的调用层次结构窗口能看到被调用方法的层次结构，双击方法名称，可以立即定位到该方法定义的地方。

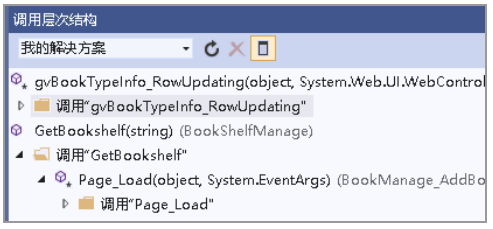

图 1-30　调用层次结构窗口

3. 突出显示引用

选中任何一个符号，如方法、属性、变量等，在如图 1-31 所示的代码编辑器中将自动突出显示此符号的所有实例。用户还可以通过快捷键“Ctrl+Shift+向上/向下键”从一个加亮的符号跳转到下一个加亮的符号。

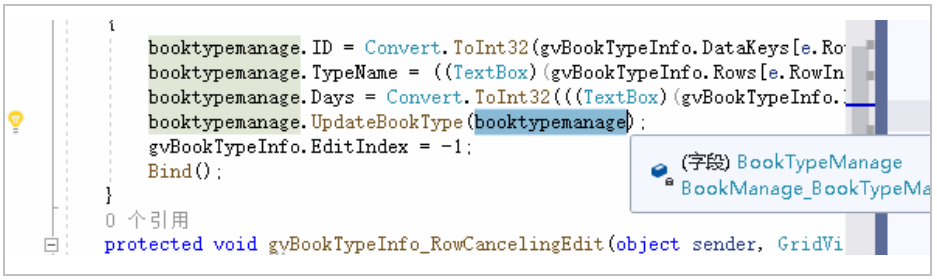

图 1-31　突出显示引用

4. 智能感知

在 Visual Studio 2019 中智能感知(IntelliSense)功能又进行了完善和加强，在输入一些关键字

时，其搜索过滤功能并不只是将关键字作为查询项的开头，而是包含查询项所有位置。有时需要使用 switch、foreach、for 等类似语法结构，只需加入语法关键字，并按两下 Tab 键，Visual Studio 2019 就会自动完成相应的语法结构。这一功能大大提高了开发人员的编程效率。

1.5 配置 ASP.NET 应用程序

在 ASP.NET 应用程序中，用户可以在系统提供的配置文件 Web.config 中对该应用程序进行配置，可以配置的信息包括错误信息显示方式、会话存储方式和安全设置等。

Web.config 文件是一个 XML 文本文件，它用来储存 ASP.NET Web 应用程序的配置信息(如最常用的设置 ASP.NET Web 应用程序的身份验证方式等)，它可以出现在应用程序的每一个目录中。

通过 Visual Studio 新建一个 Web 应用程序后，默认情况下会在根目录自动创建一个默认的 Web.config 文件。由于 ASP.NET 的 Machine.config 文件自动注册所有的 ASP.NET 标识、处理器和模块，所以在 Visual Studio 中创建新的空白 ASP.NET 项目时，会发现默认的 Web.config 文件，此文件既干净又简洁，不像以前的版本有 100 多行代码。

如果要修改配置设置，可以在 Web.config 文件下的 Web.Release.config 文件中进行重新配置。它提供重写或修改 Web.config 文件中定义的设置。在运行时对 Web.config 文件的修改不需要重启服务就可以生效(注意<processModel>节例外)。此外，Web.config 文件是可以扩展的。用户可以自定义新配置参数，并编写配置节处理程序以对它们进行处理。

Web.config 配置文件的所有代码都应该位于<configuration><system.web>和</system.web></configuration>节之间。

1. <authentication>节

<authentication>节通常用来配置 ASP.NET 身份验证支持(参数可以是 Windows、Forms、PassPort、None 4 种)。该元素只能在计算机、站点或应用程序级别声明。<authentication>元素必须与<authorization>节配合使用。

例如，基于窗体的身份验证站点的配置代码如下。

```
1. <authentication mode="Forms" >
2. <forms loginUrl="Login.aspx" name=".ASPXAUTH"/>
3. </authentication>
```

代码说明：第 1 行和第 3 行定义<authentication>节，把 mode 属性设置为 Forms，表示这个站点将执行基于窗体的身份验证，第 2 行定义当没有登录身份的用户访问页面时自动跳转到的页面，其中元素 loginUrl 表示登录网页的名称，name 表示 Cookie 名称。

2. <authorization>节

<authorization>节通常用来控制对 URL 资源的客户端访问(如允许匿名读者访问)。此元素可以在任何级别(计算机、站点、应用程序、子目录或页)上声明。<authorization>元素必须与<authentication>节配合使用。可以使用 user.identity.name 来获取已经过验证的当前的用户名；

也可以使用 web.Security.FormsAuthentication.RedirectFromLoginPage 方法将已验证的用户重定向到刚才请求的页面。

例如，禁止匿名用户访问的站点的配置代码如下。

```
1. <authorization>
2. <deny users="?"/>
3. </authorization>
```

代码说明：第 1 行和第 3 行定义<authorization>节，第 2 行通过设置<deny users="?"/>来实现任何来访的用户都需要身份认证。

3. <compilation>节

<compilation>节通常用来配置 ASP.NET 使用的所有编译设置。debug 属性的默认值为 True。在程序编译完成交付使用之后应将其设为 True。

4. <customErrors>节

<customErrors>节通常用来为 ASP.NET 应用程序提供有关自定义错误信息。但它不适用于 XML Web services 中发生的错误。

例如，当发生错误时，将网页跳转到自定义的错误页面的配置代码如下。

```
1. <customErrors defaultRedirect="ErrorPage.aspx" mode="RemoteOnly">
2. </customErrors>
```

代码说明：第 1 行和第 2 行定义<customErrors>节，并通过属性 defaultRedirect 来定义发生错误时跳转的页面是 ErrorPage.aspx。

5. <httpRuntime>节

<httpRuntime>节通常用来配置 ASP.NET HTTP 运行库设置。该节可以在计算机、站点、应用程序和子目录级别声明。

例如，ASP.NET HTTP 运行库设置代码如下。

```
<httpRuntime maxRequestLength="1024" executionTimeout="150" appRequestQueueLimit= 50"/>
```

代码说明：这段代码的含义是控制用户上传文件最大为 1M，最长时间为 150 秒，最多请求数为 50。

6. <pages>节

<pages>节通常用来标识特定于页的配置设置(如是否启用会话状态、视图状态，是否检测用户的输入等)。<pages>节可以在计算机、站点、应用程序和子目录级别声明。

例如，检测用户在浏览器输入的内容中是否存在潜在的危险数据的代码如下。

```
<pages buffer="true" enableViewStateMac="true" validateRequest="false"/>
```

代码说明：buffer="true"定义了页面发送前先缓冲输出。enableViewStateMac="true"表示在从客户端回发页时将检查加密的视图状态，以验证视图状态是否已在客户端被篡改。validateRequest="false"表示 ASP.NET 检查从浏览器输入的所有数据，以找出潜在的危险数据。

1.6 综合练习

下面通过本书的第一个 Web 应用程序来介绍创建 ASP.NET 应用程序的过程。本练习将实现在运行程序后，在浏览器中显示“我创建的首个 ASP.NET 网页”。

(1) 启动 Visual Studio 2019，选择“文件”|“新建项目”命令，打开“创建新项目”对话框，然后选择“ASP.NET Web 应用程序”，单击“下一步”按钮，在弹出的对话框中的“项目名称”文本框中输入“综合练习”，并在“位置”文本框中输入相应的存储路径，在“解决方案名称”文本框中输入“综合练习”。最后，单击“创建”按钮，如图 1-32 所示。

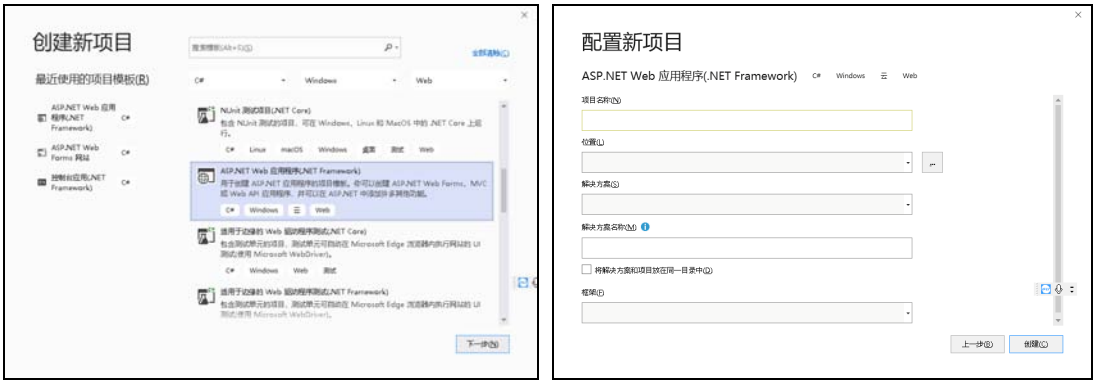

图 1-32　配置新建项目

(2) 在“解决方案管理器”中的网站根目录下会生成一个“综合练习”的 Web 项目。右击项目名称“综合练习”，在弹出的菜单中选择“添加”|“新建项”命令。

(3) 弹出“添加新项”对话框，选择“已安装”模板下的 Web 模板，并在模板文件列表中选中“Web 窗体”，然后在“名称”文本框输入该文件的名称 Default.aspx，最后单击“添加”按钮。

(4) 此时，“综合练习”目录下面会生成一个如图 1-33 所示的 Default.aspx 页面，它包括两个文件，一个是 Default.aspx.cs 文件，用于编写程序的后台代码；另一个是 Default.aspx.designer.cs 文件，存放一些页面控件中控件的配置信息。

(5) 双击网站的根目录下的 Default.aspx 文件，进入“视图设计器”窗口。从“工具箱”拖动一个“Label 控件”到如图 1-34 所示的“设计视图”中。

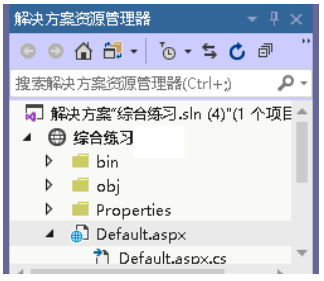

图 1-33　生成 Default.aspx 页面

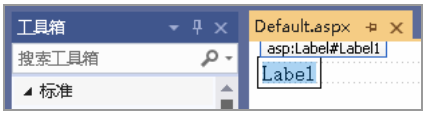

图 1-34　设计视图

(6) 双击网站根目录下的 Default.aspx.cs 文件，编写代码如下：

```
1. protected void Page_Load(object sender, EventArgs e){
2.             Label1 .Text ="我创建的首个 ASP.NET4.0 网页！";
3. }
```

代码说明：第 1 行处理页面 Page 的加载事件 Load。第 2 行设置 Label1 控件的文本显示“我创建的首个 ASP.NET4.0 网页！”。

(7) 按 Ctrl+F5 键运行程序，效果如图 1-35 所示。

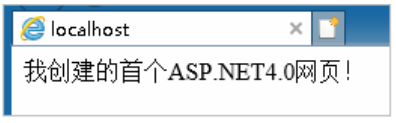

图 1-35　网页效果

1.7 习题

一、填空题

1. ASP.NET 支持的编程语言有_____、_____等。

2. .NET 基类库位于_____的上层，与_____紧密集成在一起，可被.NET 支持的任何语言所使用。

3. ASP.NET 网站在编译时，首先将语言代码编译成_____。

4. ASP.NET 页面文件的后缀是_____。

5. 基于 C#的 ASP.NET 程序文件的后缀是_____。

二、选择题

1. ASP.NET 网页是完全面向对象的。在 ASP.NET 网页中，可以使用(　　)来处理 HTML 元素。

A. 属性　　B. 方法　　C. 事件　　D. 过程

2. (　　)不属于 ASP.NET 开发和运行环境。

A. 安装 IIS　　B. SQL Server 数据库

C. 安装.NET Framework SDK　　D. Visual Studio.NET

3. .NET Framework 旨在实现的目标包括(　　)。

A. 提供一个一致的面向对象的编程环境，而无论对象代码是在本地存储和执行，还是在本地执行但在 Internet 上分布，或者是在远程执行的

B. 提供一个将软件部署和版本控制冲突最小化的代码执行环境

C. 提供一个可提高代码(包括由未知的或不完全受信任的第三方创建的代码)执行安全性的代码执行环境

D. 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境

4. HTTP 的常用请求方法包括(　　)。

A. PUT　　B. LINK　　C. DELETE　　D. UNLINK

5. .NET Framework 具有的主要组件是(　　)。

A. 公共语言运行库　　B. .NET Framework 类库

C. 动态语言运行时　　D. 中间语言

三、上机题

1. 使用记事本编写一个 HTML 的静态网页，运行后在网页中显示“欢迎来到 ASP.NET 的世界”。

2. 在本地电脑中安装 IIS Web 服务器并进行相应的配置，然后创建一个名为 MyRoot 的虚拟目录。

3. 在本地电脑中安装 Visual Studio 2019 专业版开发环境，并熟悉开发主界面和菜单栏的选项。

4. 创建一个 Web 网站，运行后浏览器页面中显示 Welcome to My Website！。