高等院校计算机应用系列教材

ASP.NET Web开发技术（微课版）

王　颖　刘　艳　王先水　主　编

清華大學出版社
北　京

内 容 简 介

ASP.NET Web 是目前软件开发市场比较流行的一种开发技术，可配合任何一种.NET 平台下的语言进行开发。本书以构建 SPOC 混合教学模式对 ASP.NET Web 开发技术课程进行总体设计：课程从“准职业人”的角度，以工作过程为导向、工作任务为基础、学生能力为落脚点，突出培养学生软件设计、代码编写、算法设计能力，通过课内、课外双线同步并实施教学。

本书共 10 章，主要内容包括 Web 技术概述、ASP.NET Web 标准服务器控件、用户控件和母版页技术、站点导航控件、ASP.NET 常用内置对象与数据传递、ASP.NET 状态管理、ADO.NET 数据库访问技术、数据绑定与数据绑定控件、ASP.NET AJAX 控件、三层架构和 MVC 开发技术等。书中的所有案例均来自编者多年的教学手稿笔记及项目开发经验，具有一定的实用性。

本书可作为高等院校计算机相关专业的 Web 开发、网络程序设计、Web 数据库应用技术等课程的教材，也可作为对 Web 应用开发有兴趣的人员的自学用书。

图书在版编目(CIP)数据

ASP.NET Web 开发技术：微课版 / 王颖，刘艳，王先水主编. —北京：清华大学出版社，2023.1
高等院校计算机应用系列教材
ISBN 978-7-302-62103-4

I. ①A… II. ①王… ②刘… ③王… 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(2022)第 198320 号

责任编辑：刘金喜
封面设计：高娟妮
版式设计：妙思品位
责任校对：成凤进
责任印制：刘海龙

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：010-83470000　　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：三河市龙大印装有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　印　　张：19.5　　字　　数：438 千字
版　　次：2023 年 1 月第 1 版　　印　　次：2023 年 1 月第 1 次印刷
定　　价：79.00 元

产品编号：099238-01

前　言

ASP.NET Web 是目前软件开发市场比较流行的一种开发技术。该技术易学易用，开发效率高，可配合任何一种.NET 平台下的语言进行开发。

本书以构建 SPOC 混合教学模式对 ASP.NET Web 开发技术课程进行总体设计：课程从“准职业人”的角度，以工作过程为导向、工作任务为基础、学生能力为落脚点，突出培养学生软件设计、代码编写、算法设计能力，通过课内、课外双线同步并实施教学，培养 Web 开发设计、数据库设计、ASP.NET 后台开发技术、ADO.NET 技术等方面的高技能与高素质应用型人才；按职业岗位能力设计 Web 网站设计基础模块、Web 开发控件基础模块、ADO.NET 数据库访问技术模块、ASP.NET 项目上机实验模块四大课程模块。

本书的目的在于让广大学生和学员更快、更好地理解和掌握 ASP.NET Web 开发技术的每个知识要点。本书在整理时参考了目前市面上已有的相关书籍，集各家之所长，结合多年的教学手稿笔记进行扩展与整理，将一些原本深奥并难以理解的开发技术思想通过一些简单的案例进行解析，让学生能够轻松掌握开发技术思想的精髓。

本书以“案例驱动教学”为整体编写原则，所有开发技术知识要点均基于一两个案例，通过案例来加深对项目开发技术思想的理解。书中设计的案例来自企业真实项目的拆分，上机实验部分体现了知识的综合应用及设计开发能力的培养。书中的例题、上机实验内容均在 Visual Studio 2013 以上版本开发平台下通过测试且运行无误。在这种思想指导下，组织本书的内容如下。

第 1 章　Web 技术概述：主要介绍软件体系架构、Web 工程原理、Web 网页开发技术，以及 ASP.NET 的运行、开发环境和配置。

第 2 章　ASP.NET Web 标准服务器控件：主要介绍常用的 ASP.NET Web 标准服务器控件的属性、事件、方法及应用，ASP.NET 验证控件的基本属性、事件、方法及应用。

第 3 章　用户控件和母版页技术：主要从项目开发要求页面风格的一致角度出发，讲解用户控件的创建和使用、母版页的创建和使用。

第 4 章　站点导航控件：主要介绍站点导航控件在网站开发中的基本应用，以及指示当前操作页面的具体位置。

第 5 章　ASP.NET 常用内置对象与数据传递：主要介绍 ASP.NET 各内置对象的属性、方法及基本应用，跨页传递数据的基本方法和应用。

第 5 章 ASP.NET 常用内置对象与数据传递：主要介绍 ASP.NET 各内置对象的属性、方法及基本应用，跨页传递数据的基本方法和应用。

第 6 章 ASP.NET 状态管理：主要介绍 Cookie 对象、Session 对象和 Application 对象的常用属性及基本应用。

第 7 章 ADO.NET 数据库访问技术：主要介绍 ADO.NET 数据模型及常用对象、数据库的连接字符串、连接数据库的 Connection 对象、执行数据库命令的 Command 对象、读取数据的 DataReader 对象、DataSet 对象、数据读取器的 DataAdapter 对象。

第 8 章数据绑定与数据绑定控件：主要介绍数据绑定表达式实现数据绑定的基本原理，常用数据绑定控件 GridView、DetailsView、FormView 等的常用属性，以及绑定数据的基本原理和基本应用。

第 9 章 ASP.NET AJAX 控件：本章主要实现用户体验，重点介绍 AJAX 技术工件原理、常用 ASP.NET AJAX 控件的基本语法和基本应用。

第 10 章三层架构和 MVC 开发技术：简单介绍三层架构的基本原理和 MVC 开发技术的入门知识。

为方便教师教学和学生自学，《ASP.NET Web 开发技术(微课版)》对重点知识和案例通过嵌入二维码的形式进行视频讲解，同时配套相应的免费讲稿、PPT 课件、教学大纲、实验大纲和案例源代码，这些资源可通过扫描下方二维码下载。

教学资源

《ASP.NET Web 开发技术(微课版)》在武汉轻工大学机械工程学院和武汉工程科技学院信息工程学院的大力支持下，由武汉轻工大学机械工程学院材料成型系王颖、武汉工程科技学院信息工程学院计算机系刘艳、王先水三位老师共同编写完成。书中案例全部来自教师多年上课的手稿笔记和讲稿，同时引用了参考文献中列举的 ASP.NET Web 开发技术相关书籍中的部分内容，吸取了同行的宝贵经验，在此谨表谢意。

因编者水平有限，书中难免有不当之处，欢迎广大读者批评指正。

服务邮箱：476371891@88.com。

编者

2022 年 10 月于武汉

目　录

第 1 章

Web 技术概述

Web 技术的飞速发展增强了人们对现实世界的认识，为人们的生活提供了极大的便利。ASP.NET是进行 Web 应用程序开发的主流技术之一，该技术易学易用、开发效率高，可以配合任何一种.NET 语言进行 Web 开发。

Web 技术概论

1.1 软件体系架构

架构设计出现的背景是需要进行超越算法和数据结构的设计，以适应软件规模和复杂性的增长。由美国 Borland 公司最早研发的 C/S(client/server，客户/服务器)架构技术、美国微软公司研发的 B/S(browser/server，浏览器/服务器)架构技术是当今应用软件开发领域在主流软件中的开发体系结构。

1.1.1 C/S 架构

C/S 架构是典型的两层架构，其客户端包含一个或多个在用户计算机上运行的程序。服务器端有两种：一种是数据库服务器端，客户端通过数据库连接访问服务器端的数据；另一种是 Socket 服务器端，服务器端的程序通过 Socket 与客户端的程序通信。

在 C/S 架构中，客户端需要实现绝大多数的业务逻辑和页面展示，通过与数据库的交互(通常是 SQL 或存储过程)实现持久化数据，以此满足实际项目的需要。

C/S 架构的优点：界面和操作丰富；安全性能容易保证，容易实现多层论证；只有一层交互，响应速度非常快。

C/S 架构的缺点：适用面窄，常用于局域网中；用户群固定，程序需要安装才可使用；维护成本高，软件每升级一次，其所涉及的所有客户端程序都需要升级。

1.1.2 B/S 架构

B/S 架构是三层架构，由浏览器端、Web 服务器端、数据库端构成。其中，浏览器指的是 Web 浏览器，其极少数业务逻辑在前端实现，主要的业务逻辑在服务器端实现。B/S 架构系统只需有 Web 浏览器就可访问，不存在软件安装。

B/S 架构中的显示逻辑在 Web 浏览器上完成，而业务逻辑则在 Web 服务器上完成，减少了客户端压力。

B/S 架构的优点：客户端无须安装，只需 Web 浏览器即可；B/S 架构可直接连在局域网上，通过一定的权限控制实现多客户访问的目的，交互性强；B/S 架构只需升级服务器软件。

B/S 架构的缺点：在跨浏览器方面，B/S 架构不尽人意；在速度和安全性方面需要花费巨大的设计成本。

1.2 Web 工程原理

当用户通过浏览器访问 Web 站点时，数据必须遵从 HTTP(hyper text transfer protocol，超文本传输协议)进行数据传输。HTTP 是基于 B/S 架构使用 TCP 连接在应用层进行可靠的数据传输。

按 HTTP 传输的数据实质是 W3C 网页文件。该网页文件是用 HTML(hyper text markup language，超文本标记语言)编写的可在 WWW 上传输能被浏览器识别显示的文本文件，文件的扩展名为 htm 或 html 等。

1.2.1 HTTP

因特网能够迅速发展的原因是 WWW 的迅速发展，而 WWW 不断成功的最主要原因是超文本传输协议的高效性。HTTP 是以 TCP/IP(transmission control protocol/Internet protocol，传输控制协议/网际协议)通信协议为基础，提供在 WWW 服务器和客户端浏览器之间传输信息机制，规定客户端浏览器和服务器间的交互规则。

因特网上的每个网页都具有一个唯一的名称标识，通常称为 URL(uniform resource locator，统一资源定位地址)，其可以是本机磁盘，也可以是局域网上的某一台计算机，但更多的是因特网上的网站。简单地说，URL 是 WWW 地址，即通常所说的网址，其基本格式为

传输协议：// 主机名:端口号/路径

例如：

1. HTTP 的工作原理

WWW 使用 HTTP 传输各种超文本网页和数据，HTTP 会话过程包括以下 4 个步骤。

1) 建立连接

客户端的浏览器向服务器端发出建立连接的请求，服务器端给出响应后即可建立连接。

2) 发出请求

客户端按照协议要求通过连接向服务器端发出自己的请求，客户端常采用 HTTP URL 向服务器端发出访问请求。

3) 给出应答

服务器端按照协议的要求给出应答，把结果(HTML 文件)返回给客户端。也就是说，服务器端将选中的 HTML 文档通过连接传输到客户端，并将其在浏览器中显示出来。

4) 关闭连接

客户端接到响应后关闭连接。也就是说，HTML 文件传到客户端后，服务器将会自动终止该 TCP/IP 连接。

例如，某客户端发出请求的 URL 为 http://www.wuhues.com/jigoushezhi/xueyuanshezhi.htm，假设 xueyuanshezhi 网页是一个基本的 HTML 文件，请分析完整的会话过程。客户端浏览器与服务器主机 www.wuhues.com 的会话过程如下。

(1) 客户端初始化一个与服务器主机 www.wuhues.com 中服务器的 TCP 连接，服务器使用默认端口号 80 监听来自客户端的连接建立请求。

(2) 客户端与 TCP 连接相关的本地套接字发出一个 HTTP 请求消息，该消息中包含路径 jigoushezhi/xueyuanshezhi.htm。

(3) 服务器与 TCP 连接相关联的本地套接字接收这个请求消息，再从服务器主机的内存或磁盘中取出对象 jigoushezhi/xueyuanshezhi.htm，经由同一个套接字发出包含该对象的响应消息。

(4) 服务器告知 TCP 关闭该 TCP 连接，但TCP 要到客户端收到刚才的响应消息之后才会真正终止该连接。

(5) 客户端由同一个套接字接收该响应消息，TCP连接随后终止，该消息表明所封装的对象是一个 HTML 文件。

2. HTTP 的特点

HTTP 的特点主要有：以 B/S 架构为基础；简单快速，浏览器向服务器请求服务时只需传送请求方法和路径，HTTP 简单，使得服务器和程序规模小，通信速度快；HTTP 允许传输任意类型的数据对象；浏览器向服务器发出一个请求，服务器响应处理客户的请求并收到客户的应答后，断开服务器与浏览器的连接，从而节约传输时间；HTTP对于事务处理没有记忆能力。

1.2.2　网页开发技术

网页分为静态网页和动态网页两大类，相应的网页开发技术也分为静态网页开发技术和动态网页开发技术。

1. 静态网页开发技术

静态网页是指用纯 HTML代码编写的网页，并保存扩展名为 html 或 htm 的文件形式。这种用纯 HTML代码编写的网页在设计完成后，任何人在任何时候采用任何方式浏览该页面，其效果都是相同的。因此，这种网页的内容更新较烦琐，必须是设计制作好之后用专门的软件上传到服务器上才能更新。

静态网页开发技术适用于更新较少的展示型网站，可用于传统的媒体广告，会出现各种动态效果，如 GIF 格式的动画、滚动字幕等，但这些动态效果只是视觉上的。

静态网页的执行过程：用户通过客户端浏览器输入网址并按 Enter 键后，发出 WWW 请求；服务器收到静态网页请求；服务器从硬盘的指定位置查找相应的 HTML 文件；将硬盘中找到的 HTML 文件返回给服务器；服务器向客户端返回请求的文件；客户端浏览器收到请求的文件，解析这些 HTML 代码并将其显示出来。

静态网页中没有程序代码，只有HTML 标记，但静态网页中可以包含客户端脚本，常见的客户端脚本语言有 JavaScript。客户端脚本在一个特定的网页中改变界面及行为，或者响应鼠标或键盘操作。也就是说，静态网页的动态行为都是在客户端进行的，而网页是静态生成的。

2. 动态网页开发技术

动态网页是执行时用户可以输入允许的各种信息，以实现人机交互，能根据不同的时间、不同的访问者显示不同的内容。采用动态网页开发技术可实现用户注册、用户登录、用户管理、订单管理等操作。动态网页的文件格式根据不同的程序设计语言而定，常见的有 ASP、JSP、PHP 等。

动态网页的执行过程：用户通过客户端浏览器输入网址并按 Enter 键后，发出 WWW 请求；服务器收到动态网页请求；服务器从硬盘的指定位置查找相应的动态网页文件；将硬盘中找到的动态网页文件返回给服务器；服务器执行其中的程序代码，生成 HTML 文件；服务器向客户端返回 HTML 文件，客户端浏览器收到请求的文件，并以图形方式将 HTML 标记显示在浏览器上。

对于动态网页，服务器的主要功能是通过文件系统找到用户要访问的动态网页文件，执行该网页文件中的程序代码，产生 HTML 文件，再将 HTML 文件传给客户端浏览器。

1.3 Web 网页开发技术

WWW 是一种典型的分布式应用架构，其应用中的每次信息交换都涉及客户端和服务器端两个层面，因此 Web 网页开发技术主要分为 Web 客户端技术和 Web 服务器端技术。

1.3.1 Web 客户端技术

Web客户端的首要任务是展现信息内容，而 HTML 是信息展现的最有效的载体之一。最初的 HTML 只能在浏览器中展现静态的文本或图像信息，满足不了人们对信息丰富和多样性的强烈要求，因此由静态网页技术向动态网页技术转变成为 Web 客户端技术演进的必然趋势。目前，支持 Web 客户端动态技术的语言主要有 VBScript、JavaScript 脚本语言。

1. HTML

HTML是 Internet 用于编写网页的主要语言，它提供了精简而有力的文件定义，可以设计出多姿多彩的超媒体文件。借助 HTTP，HTML 文件可以在全球的互联网上进行跨平台的文件交换。

HTML 文件是纯文本的文件格式，文件中的文字、字体、字号大小、段落、图片、表格及超链接，甚至文件名称都是用不同意义的标签来描述的，以此来定义文件的结构与文件间的逻辑关联。简而言之，HTML 是以标签来描述文件中的多媒体信息。

客户端浏览器按顺序解释执行 HTML 文件，对 HTML 文件中的标签用错或属性用错，既不报错，也不终止执行。不同的浏览器对同一标记会有不同的解释，也就有不同的显示效果。

2. CSS

CSS(cascading style sheet，样式表)属于动态 HTML 技术，扩充了 HTML 标记的属性设置，使得显示页面效果更加丰富，表现效果更加灵活，它与 DIV 配合使用可以很好地对页面进行分割和布局。CSS 对页面元素、布局更加精确，同时能够实现页面内容与表现形式的分离，使得网站的设计风格趋向统一，维护更加容易。

3. JavaScript 脚本语言

JavaScript 是一种轻量级的直译式编程语言，基于 ECMAScript 标准(一种由 ECMA 国际组织，通过 ECMA-262 标准化的脚本程序语言)，于 1995 年由网景(Netscape)公司开发，基于对象的、采用事件驱动的脚本语言。JavaScript、HTML 和 CSS 被称为“Web 前端开发的三大技术”，目前 JavaScript 已经广泛应用于 Web 开发，当今几乎所有的浏览器都支持 JavaScript，无须额外安装第三方插件。

JavaScript 脚本语言的特点：JavaScript 是一种直译式的脚本语言，无须事先编译，要以在程序运行的过程中逐行解释的方式使用；JavaScript 具有非常简单的语法，无须定义变量类型，所有变量的声明都可以用统一的类型关键字表示；JavaScript 语言是一种 Web 程序开发语言，只与浏览器的支持有关，与操作系统的平台类型无关；JavaScript 语言对大小写非常敏感。

1.3.2　Web 服务器端技术

Web 服务器端技术是指服务器端的脚本编程技术，常用的服务器脚本语言有 CGI、ASP、PHP、JSP、ASP.NET、Python 等。这些脚本语言的共同特点是都运行于服务器端，能够动态生成网页，脚本运行不受客户端浏览器的限制，脚本程序都是将脚本语言嵌入 HTML 文档中，执行后返回给客户端的是 HTML 代码。

1. CGI

CGI 即通用网关接口，是一种早期的动态网页技术，可以使用不同的程序设计语言编写适合的 CGI 程序，如 VB、C/C++等。该技术已经发展成熟且功能强大，但因编程困难，效率低下，修改复杂，所以逐渐被新技术取代。

2. ASP

ASP 是 Microsoft 开发的服务器脚本环境，内置于 IIS 3.0 及以后的版本中，通过 ASP

可结合 HTML 网页、ASP 指令和 ActiveX 组件建立动态、交互且高效的 Web 服务器应用程序。ASP 动态网页技术开发网页程序代码是在服务器端执行的，服务器端将程序执行的结果返回给客户端浏览器，从而减轻客户端浏览器的负担，极大提高了服务器端与客户端浏览器的交互速度。

3. PHP

PHP 是一种易学习和使用的服务器端脚本语言，用户只需要很少的编程知识就能使用 PHP 建立一个真正交互的 Web 网站。PHP 不需要特殊的开发环境，不仅支持多种数据库，还支持多种通信协议。

4. JSP

JSP 是由 Sun 公司推出的，基于 Java Servlet 及 Java 体系的 Web 开发技术。在 HTML 文档中嵌入 Java 程序片段(Scriptlet)和 JSP 标记可形成 JSP 文件。JSP 脚本运行于服务器端，可以跨 UNIX、Linux、Windows 平台使用。JSP 代码需编译成 Servlet 并由 Java 虚拟机执行，编译操作仅在对 JSP 页面第一次请求时发生。

5. ASP.NET

ASP.NET 是继 ASP 后推出的全新动态网页开发技术，是建立在.NET Framework 公共语言运行库上的，可用于在服务器上生成功能强大的 Web 应用程序。

6. Python

Python 是由荷兰人 Guido van Rossum 发明，于 1991 年公开发行的一种脚本语言。它是面向对象的解释型程序设计语言，具有简洁性、易读性和可扩展性，广泛应用于系统管理任务的处理和 Web 的开发。

1.4 ASP.NET 基础知识

ASP.NET 是 Microsoft 公司的 Web 服务器端脚本技术，运行在 IIS(internet information server，互联网信息服务)服务器中的程序，可以使嵌入网页中的脚本由 WWW 服务器执行，其是动态网页技术，也是面向新一代企业级的网络计算 Web 平台，还是.NET Framework (.NET 框架)中的一部分，可使用.NET 兼容的语言编写 ASP.NET 应用程序。

在 ASP.NET 网页中，可以使用 ASP.NET 服务器端控件建立常用的用户接口元素，并对其进行编程；还可以使用内建可重用组件和自定义组件快速建立 Web 网页，从而使代码极大简化。

1.4.1 ASP.NET 引擎

在处理动态网页时，服务器既要查找动态网页文件，同时还要执行动态网页文件以生成 HTML 文件，为了减轻服务器的压力，可将 Web 服务器和动态网页源代码的执行分离开来。当一个 Web 请求到达时，Web 服务器判断所请求的页面是静态网页还是动态网页，如

果是静态网页，则服务器直接将该页面内容发送到请求浏览器；如果是动态网页，如一个 ASP.NET 网页，则服务器把执行该网页的任务转交给 ASP.NET 引擎，由 ASP.NET 引擎执行动态网页中的程序代码，以 HTML 文件形式返回给 Web 服务器，再由 Web 服务器将该 HTML 文件发送到请求的客户机浏览器。

ASP.NET 动态网页的请求过程如图 1.1 所示。图 1.1 中常见的配置有：客户机浏览器，安装有 IE 浏览器；Web 服务器，配置有 IIS；数据库服务器，安装有 SQL Server 数据库管理系统。

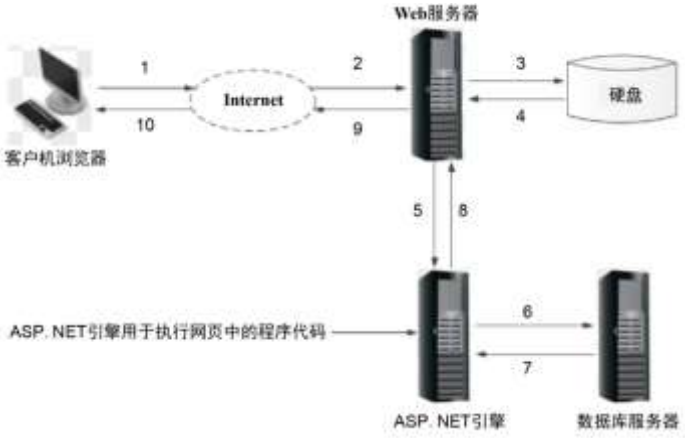

图 1.1　ASP.NET 动态网页请求过程

图 1.1 的 ASP.NET 动态网页请求过程如下。

(1) 客户端通过浏览器发出 Web 请求。

(2) Web 服务器收到 ASP.NET 动态网页请求。

(3) Web 服务器从硬盘的指定位置查找相应的 ASP.NET 动态网页文件。

(4) 将在硬盘中找到的 ASP.NET 动态网页文件返回给 Web 服务器。

(5) Web 服务器将 ASP.NET 动态网页文件发送给 ASP.NET 引擎。

(6) ASP.NET 引擎逐行读取该文件，并执行程序中的代码，如果需要访问数据库，则将这部分代码发给数据库服务器；如果不需要访问数据库，则直接转到步骤(8)。

(7) 数据库服务器执行数据库访问，并将结果返回给 ASP.NET 引擎。

(8) ASP.NET 引擎生成最终的纯 HTML 文件并返回给 Web 服务器。

(9) Web 服务器将纯 HTML 文件发送给客户端浏览器。

(10) 客户端收到请求的纯 HTML 文件，并在浏览器中以图形方式将 HTML 标记显示在计算机屏幕上。

由于网页处理发生在 Web 服务器上，因此网页执行的每个操作都需要一次到服务器的往返过程。ASP.NET 网页可以执行客户端脚本，而客户端脚本不需要到服务器的往返过程，这对于客户输入数据验证和某些类型的用户界面编程十分有用。

1.4.2　ASP.NET 应用程序开发工具

ASP.NET 应用程序开发模式有 ASP.NET Web 窗体、ASP.NET MVC、ASP.NET Core 等，

实际开发时根据具体需求和公司开发人员技术背景而定。本书采用 ASP.NET Web 窗体开发模式。

开发 ASP.NET 应用程序的主要开发工具是 Visual Studio，本书采用 Visual Studio 2013。

在开发 ASP.NET Web 窗体程序时，首要任务是开发 ASP.NET 动态网页，而执行网页的程序代码软件是 ASP.NET 引擎，在安装 Visual Studio 版本后，计算机自动配置好 ASP.NET 引擎。

Visual Studio 是 Microsoft 推出的用于软件开发的重要平台，目前最高版本为 Visual Studio 2021，内置.NET Framework 4.7 及以上版本。Visual Studio 开发平台将程序设计中需要的各个环节(如界面设计、程序设计、运行和调试程序)集成在同一个窗口中，极大地方便了开发人员的设计工作，通常将这种集多项功能于一体的开发平台称为集成开发环境(integrated development environment，IDE)。

1．安装 Visual Studio 开发环境

Microsoft Visual Studio 2013 安装程序通常自带.NET Framework 4.5，读者也可从微软官方网站下载。

Microsoft Visual Studio 2013 安装过程如图 1.2 所示。

图 1.2　Microsoft Visual Studio 2013 安装过程

2．安装和配置 Web 服务器

IIS 是 ASP.NET 唯一可以使用的 Web 服务器，目前常用的版本是 IIS 7.0，下面以 Windows 11 配置进行说明。

(1) 在“控制面板”的程序中选择“启用或关闭 Windows 功能”，这是一个触发 UAC 的操作，如果 Windows 11 没有关闭 UAC，则会弹出提示信息，确认并继续。

(2) 如果仅需要IIS 7.0 支持静态内容，可直接选中“Internet 信息服务”，如果希望支持动态内容，则需要展开“万维网服务”分支，将所需要的选项全部选中。

(3) 单击“确定”按钮，Windows 11 即启动 IIS 的安装过程。

1.4.3　.NET Framework 体系结构

.NET 平台是 Microsoft 公司 2000 年推出的全新的应用程序开发平台，可用来构建和运行新一代 Microsoft Windows 和 Web 应用程序。它建立在开放体系结构之上，集 Microsoft 在软件领域的主要技术成就于一身。.NET 框架也是 Windows 7、Windows 8、Windows 10 操作系统的核心部件。

.NET 平台主要由.NET Framework(.NET 框架)、基于.NET 的编程语言、开发工具 Visual Studio 构成。其中.NET Framework 是基础和核心。.NET Framework 体系结构如图 1.3 所示。

图 1.3　.NET Framework 体系结构

1. CLS

.NET Framework 中定义了一个 CLS(common language specification，公共语言运行规范)，包含函数调用方式、参数传递方式、数据类型和异常处理方式等。在进行程序设计时，如果使用符合 CLS 的开发语言(称为.NET 兼容语言，如 C#语言、VB 语言等)，那么所开发的程序可以在任何公共语言开发环境的操作系统下执行。

2. ADO.NET 和 XML

.NET Framework 直接支持 ADO.NET(数据访问接口)和 XML 文件的操作。在 XML 文档和数据集之间可以进行数据转换，甚至共享一份数据，程序员可以选择熟悉的方式处理数据，提高程序设计效率。

3. .NET 类库

.NET Framework 提供了一个巨大的统一类库，该类库提供了程序员在开发程序时所需要的大部分功能，而且可以使用任何一种.NET 兼容语言加以引用。

4. CLR

在.NET Framework 下，所有的.NET 兼容语言将使用统一的虚拟机，而 CLR(common language runtime，公共语言运行库)将是所有的兼容.NET 语言在执行时所必备的运行环境，这种统一的虚拟机与运行环境可以达到跨平台的目标。

1.5 ASP.NET 的开发模式

ASP.NET 有两种开发模式，分别是 Web Forms 模式和 MVC 模式。

1.5.1 Web Forms 模式

Web Forms 模式是传统的 ASP.NET 编程模式，它是集 HTML、服务器控件和服务器代码事件驱动于一体的开发模型。Web Forms 在服务器上编译和执行，再由服务器生成 HTML 显示为网页。Web Forms 开发模式有数以百计的 Web 控件和 Web 组件，为软件开发的页面设计提高效率。

Web Forms 模式的工作原理：浏览器客户端向服务器发送窗体页面请求，服务器根据相应的后台代码文件进行业务逻辑处理，包括对数据库服务器的访问，找到浏览器请求的窗体页面文件回传给浏览器并显示其内容。

本书主要讲解 Web Forms 开发模式。

1.5.2 MVC 模式

MVC(model view controller)是模型(model)、视图(view)、控制器(controller)的缩写，是一种程序架构模式，它强制性地使应用程序的输入、处理和输出分开。MVC 将应用程序分成视图、模型、控制器 3 个核心部件，它们各自处理自己的任务。

(1) 视图。视图是用户看到并与之交互的界面。对 Web 应用程序来说，视图就是由 HTML 元素组成的界面。

(2) 模型。模型表示企业数据和业务规则，在 MVC 的 3 个部件中，模型拥有最多的处理任务。

(3) 控制器。控制器接收用户的输入并调用模型和视图去完成用户的需求，当单击页面中的超链接和发送 HTML 表单时，请求首先被控制器捕获。控制器本身不输出任何信息和做任何业务处理，它只是接收请求并决定调用哪个模型部件去处理请求，然后再确定用哪个视图来显示返回的数据。

1.6 ASP.NET Web 项目的创建

创建 ASP.NET Web 项目有两种方式：一种是创建 ASP.NET Web 应用程序项目；另一种是创建 ASP.NET Web 网站。下面分别介绍这两种方式的创建过程。

1.6.1 创建 ASP.NET Web 应用程序项目

【例题 1.1】创建一个简单的 ASP.NET Web 应用程序项目，项目名称为 Capter1_1。ASP.NET Web 应用程序项目 Capter1_1 效果如图 1.4 所示。

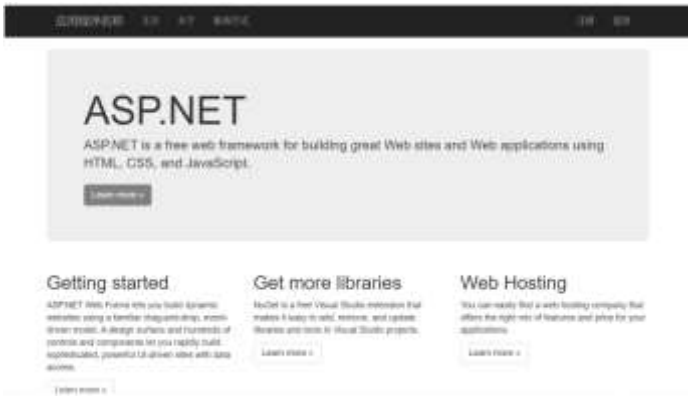

图 1.4　ASP.NET Web 应用程序项目 Capter1_1 效果

实现步骤：

1) 新建一个 ASP.NET Web 应用程序项目

(1) 启动 Visual Studio 2013，在菜单栏中单击“文件”|“项目”选项，在打开的“新建项目”对话框中，单击左侧面板中的 Visual C#下的 Web 选项，接着单击中间面板中的“ASP.NET Web”应用程序，在名称框中输入项目名称(Capter1_1)，在位置框中选择项目保存的位置，单击“确定”按钮，弹出“新建 ASP.NET 项目”对话框。

(2) 在“新建 ASP.NET 项目”对话框中，选择开发应用程序项目的类型，即 Web Forms 模板。

(3) 如果想要更改模板默认使用的身份验证方式，则可在“新建 ASP.NET 项目”对话框中单击“更改身份验证”按钮，打开“更改身份验证”对话框，在其中选择需要的身份验证方式。

(4) 单击“更改身份验证”对话框中的 OK 按钮，返回“新建 ASP.NET 项目”对话框，单击“确定”按钮，完成新建 ASP.NET Web 应用程序项目的创建，新建的 Capter1_1 项目目录结构如图 1.5 所示。

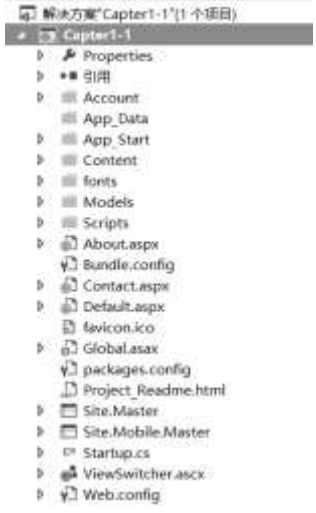

图 1.5　新建的 Capter1_1 项目目录结构

2) 编写 ASP.NET 应用程序

使用 Web Forms 模板创建 ASP.NET Web 应用程序时，Visual Studio 会自动创建一个名为 Default.aspx 的窗体页，可使用该页作为项目的主页，也就是启动页面。用户也可根据自己的需要创建新页面。

(1) 将新页面添加到项目。

在解决方案资源管理器中，右击项目名称 Capter1_1，在弹出的快捷菜单中执行“添加”|“新建项”命令，打开“添加新项”对话框，在对话框模板中间列表中选择“Web 窗体”，在“名称”文本框中输入页面名称 WebMessage，单击“添加”按钮，Visual Studio 将创建一个新页面打开其前台页面。

Visual Studio 为添加的 WebMessage 页面创建 3 个文件，分别是前台页面文件 WebMessage.aspx、后台代码文件 WebMessage.aspx.cs、设计器文件 designer.cs。页面文件中使用控件对象会在设计器文件中自动生成声明。

(2) 前台页面设计。

前台页面设计分为代码设计和视图设计两种，本书重点讲解代码设计的实现过程。ASP.NET Web 页面文件类似 HTML 文件，可以包含 HTML、XML 及脚本，ASP.NET 网页文件的脚本在服务器上运行。前台页面设计代码如下。

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebMessage.aspx.cs"
Inherits="WebProject1.WebMessage" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
    <p>欢迎来到 ASP.NET 编程世界！</p>
    <p>请输入你的姓名：<asp:TextBox ID="txtName" runat="server"></asp:TextBox></p>
    <p>
        <asp:Button ID="btnSumit" runat="server" Text="提 交"     Width="120px"/></p>
        <p>
            <asp:Label ID="lblMessage" runat="server" Text="Label"></asp:Label></p>
    </div>
    </form>
</body>
</html>
```

前台页面运行效果如图 1.6 所示。

图 1.6　前台页面运行效果

(3) 后台功能逻辑代码设计。

将源代码页面切换到视图页面，双击“提交”按钮，Visual Studio 会在前台页面的 Button 控件中增加 OnClick="btnSumit_Click"属性，同时 Visual Studio 会在编辑器的单独窗口中打开WebMessage.aspx.cs 文件，并在该文件中生成提交按钮的 Click 事件处理程序框架。后台功能逻辑代码设计如下。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebProject1
{
    public partial class WebMessage : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void btnSumit_Click(object sender, EventArgs e)
        {
            lblMessage.Text = txtName.Text + "，来到 Web 开发世界!";
        }
    }
}
```

(4) 编译运行程序。

编译ASP.NET Web应用程序后就可以运行其中包含的页面。生成应用程序并运行 Web 窗体页的方法有以下 3 种。

① 使用调试器生成并运行 Web 窗体页。

在解决方案资源管理器中，右击要运行的 Web 窗体页，在弹出的菜单中执行“设为起始页”命令。在菜单栏中单击“调试”|“启动调试”选项，或者直接按F5键。该方式为重新编译后再运行，这样可以在程序代码中通过设置断点跟踪来调试程序。

② 不使用调试器生成并运行 Web 窗体。

在解决方案资源方案管理器中，右击要运行的 Web 窗体页，在弹出的菜单中执行“设为起始页”命令。在菜单栏中单击“调试”|“开发执行(不调试)”选项，或者直接按Ctrl+F5

键。该方式直接运行生成的程序，不进行重新编译，因此运行速度较快。

③ 在浏览器中查看生成并运行 Web 窗体页。

在解决方案资源管理器中，右击要预览的 Web 窗体页，在弹出的菜单中执行“在浏览器中查看”命令，Visual Studio 会生成 ASP.NET Web 应用程序，并在默认浏览器中启动要预览的页面。

在前台页面运行的“请输入你的姓名”文本框中输入姓名，如李梦园，然后单击“提交”按钮，应用程序运行效果如图 1.7 所示。

图 1.7　应用程序运行效果

1.6.2　创建 ASP.NET Web 网站

【例题 1.2】创建一个 ASP.NET Web 网站 Capter1_2，要求在“用户名”“密码”文本框中输入相应信息，然后单击“登录”按钮，在指定标签处显示输入的信息。例题1.2 实现效果如图 1.8 所示。

图 1.8　例题 1.2 实现效果

实现步骤：

1) 新建 Web 站点

在菜单栏中单击“文件”|“网站”选项，在打开的“新建网站”对话框中，单击左侧面板中的“模板”下的 Visual C#，然后单击中间面板中的“ASP.NET 空网站”，在“Web 位置”文本框中采用默认的“文件系统”，再单击“浏览”按钮，选择网站页面，保存设定网站的名称、位置，单击“新建网站”对话框中的“确定”按钮，完成新建 Web 站点的操作。

此时创建的 Web 站点只有一个解决方案、网站根目录和配置文件，没有 ASP.NET Web 动态页面，因此需要根据需求添加相应的 Web 窗体页面。

2) 在 Web 站点中添加 Web 窗体页面

在解决方案资源管理器中右击网站根目录，在弹出的快捷菜单中依次执行“添加”|“添加新项”命令，打开“添加新项”对话框，在中间模板中选择“Web 窗体”，在名称文本框中输入页面名称 UserLogin。单击“添加”按钮，Visual Studio 将创建一个新页并将其前台页

面打开。此时 Visual Studio 将创建两个文件，分别是 UserLogin.aspx 前台页面文件和 UserLogin.aspx.cs 后台代码文件。

UserLogin.aspx 前台页面文件用户可以添加文本、服务器控件，其代码设计如下。

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <p>用户名：<asp:TextBox ID="txtName" runat="server"></asp:TextBox></p>
            <p>密　码：<asp:TextBox ID="txtPwd" runat="server" TextMode="Password">
                                    </asp:TextBox></p>
            <p><asp:Button ID="btnLogin" runat="server" Text="登录" Width="120px"
                                OnClick="btnLogin_Click" /></p>
            <p><asp:Label ID="lblShow" runat="server" Text="Label"></asp:Label></p>
        </div>
    </form>
</body>
</html>
```

UserLogin.aspx.cs 后台代码文件用户可以设计逻辑代码，其代码设计如下。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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

    }
    protected void btnLogin_Click(object sender, EventArgs e)
    {
        //假设用户名是：李梦园，密码是 2330210101
        if(txtName .Text =="李梦园"&&txtPwd .Text =="2330210101")
        {
```

```
                lblShow.Text = "登录成功！用户名：" + txtName.Text + "，密码是：" +
                                                                txtPwd.Text;
            }
            else
            {
                lblShow.Text = "登录失败，用户名或密码错误！";
            }

        }
    }
```

3) 运行 Web 窗体页程序

在解决方案资源管理器中，右击 UserLogin.aspx 页面，在弹出的快捷菜单中执行“在浏览器中查看”命令，Visual Studio 会生成 ASP.NET Web 应用程序，并在默认的浏览器中启动预览的效果，如图 1.8 所示。

1.6.3 创建 ASP.NET Web 空应用程序

【例题 1.3】创建一个 ASP.NET Web 空应用程序 Capter1_3，要求在“用户名”“密码”文本框中输入相应信息，在页面的标签处显示。例题 1.3 应用程序运行效果如图 1.9 所示。

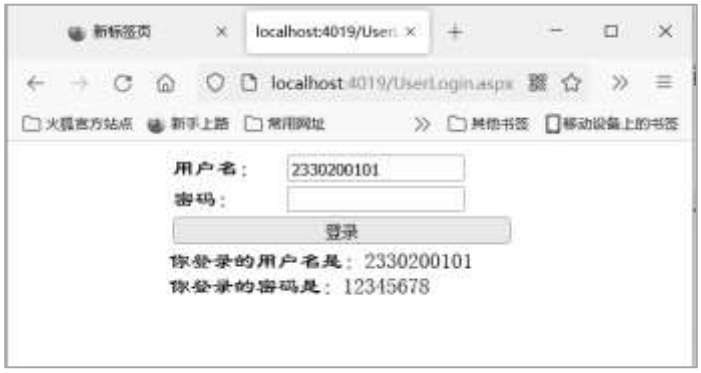

图 1.9 例题 1.3 应用程序运行效果

实现步骤：

(1) 创建 ASP.NET 空 Web 应用程序。启动 Visual Studio 2013，单击“文件”|“新建”选项，打开“新建项目”对话框，如图 1.10 所示，在该对话框的“已安装”模板中选择 Visual C# | Web | Visual Studio 2012 选项，在中间模板中选择“ASP.NET 空 Web 应用程序”选项，在“名称”文本框中输入 Capter1_3，在位置中选择项目的保存路径，单击“确定”按钮，即创建一个解决方案为 Capter1_3 下的网站根目录也为 Capter1_3 且没有任何页面的项目。

(2) 右击网站根目录 Capter1_3，在弹出的菜单中执行“添加”|“新建项”命令，打开“添加新项”对话框，如图 1.11 所示，在该对话框的中间模板中选择“Web 窗体”，在名称文本框中输入页面的名称，如 UserLogin.aspx，单击“确定”按钮，则在 Capter1_3 网站根目录下添加一个页面。

图 1.10　“新建项目”对话框

图 1.11　“添加新项”对话框

(3) UserLogin.aspx 前台页面代码设计。在页面的 DIV 层中设计一个 table 标记，在 table 标记中设计行列分别为“用户名”“密码”和“登录”按钮，同时设计 DIV 的样式，代码设计如下。

```
<body>
    <form id="form1" runat="server">
    <div style ="width :300px;margin :auto;font-family :隶书;font-size :18px">
    <table>
        <tr><td>用户名：</td><td>
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox></td></tr>
        <tr><td>密码：</td><td>
            <asp:TextBox ID="txtPwd" runat="server" TextMode ="Password">
                        </asp:TextBox></td></tr>
        <tr><td colspan ="2">
            <asp:Button ID="btnLogin" runat="server" Text="登录"    Width="280px"
                        OnClick="btnLogin_Click"/></td></tr>
    </table>
        <asp:Label ID="lblMessage" runat="server" Text=""></asp:Label>
```

```
        </div>
        </form>
    </body>
```

(4) UserLogin.aspx 页面后台“登录”按钮逻辑功能代码设计。将页面源视图切换到页面设计视图，单击“登录”按钮，进入登录后台代码设计区，代码设计如下。

```
namespace Capter1_3
{
    public partial class UserLogin : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnLogin_Click(object sender, EventArgs e)
        {
            lblMessage.Text = "你登录的用户名是：" + txtName.Text + "</br>" + "你登录的密码是："
                                                                    + txtPwd.Text;
        }
    }
}
```

(5) 运行程序。编译程序，右击“解决方案 Capter1_3”，执行“生成解决方案”或“重新生成解决方案”命令。在页面源视图中，右击执行“在浏览器中查看”命令，或者直接单击工具栏中的“调试”按钮，或者执行菜单栏中的“调试”|“启动调试”命令，或者直接按功能键 F5。程序运行的结果如图 1.9 所示。

1.7 上机实验

1. 实验目的

(1) 熟悉 ASP.NET Web 的开发环境 Visual Studio 2013。

(2) 掌握利用解决方案管理网站和创建网站的基本过程。

(3) 掌握运用静态页面设计与动态页面设计用户登录页面的基本方法。

(4) 掌握 IIS 中网站、Web 应用程序、虚拟目录创建和默认文档设置的过程。

(5) 掌握利用 Visual Studio 2013 发布 Web 应用的过程。

2. 实验内容

创建一个“Experiment1_学号姓名”解决方案，在该方案中分别添加 IndexUserLogin.html 静态页面和 IndexUserLogin.aspx 动态页面。在动态页面上实现用户名、密码的信息输入并在浏览器页面上显示出来。动态页面运行效果如图 1.12 所示。

3. 实验步骤

1) 实验内容分析

网站名称为“Experiment1_学号姓名”，在该网站下有两个页面且页面间互相独立。页

面上需要设计用户名和密码输入框及两个按钮，静态页面和动态页面所用的控件有所区别。信息如何显示可借用 C#语言课程知识通过标签来实现。

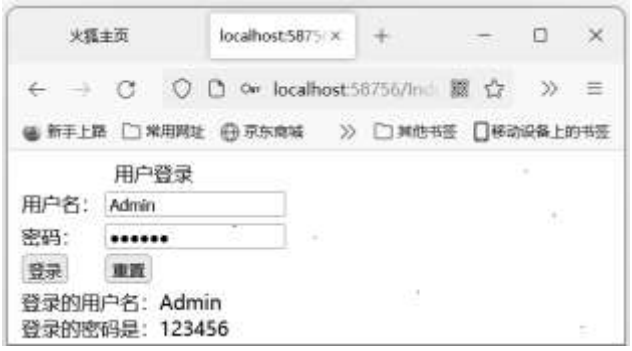

图 1.12　动态页面运行效果

2) 实验过程

启动 Visual Studio 2013，创建“Experiment1_学号姓名”解决方案，按实验内容添加页面。IndexUserLogin.html 静态页面代码设计如下。

```
<body>
    <table>
        <caption>用户登录</caption>
        <tr>
            <td>用户名：</td>
            <td><input id="txtName" type="text" /></td>
        </tr>
        <tr>
            <td>密码：</td>
            <td><input id="txtPwd" type="password" /></td>
        </tr>
        <tr>
            <td><input id="btnLogin" type="button" value="登录" /></td>
            <td><input id=" btnReset" type="reset" value="重置" /></td>
        </tr>
    </table>
</body>
```

IndexUserLogin.aspx 动态页面代码设计如下。

```
<body>
    <form id="form1" runat="server">
    <div>
     <table>
            <caption >用户登录</caption>
            <tr>
                <td>用户名：</td>
                <td> <asp:TextBox ID="txtName" runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td>密码：</td>
                <td>
```

```
                    <asp:TextBox ID="txtPwd" runat="server" TextMode
                                ="Password"></asp:TextBox></td>
            </tr>
            <tr>
                <td>
                    <asp:Button ID="btnLogin" runat="server" Text="登录"
                                OnClick="btnLogin_Click" /></td>
                <td>
                    <asp:Button ID="btnReset" runat="server" Text="重置" /></td>
            </tr>
            <tr>
                <td colspan="2">
                    <asp:Label ID="lblShow" runat="server" Text=""></asp:Label></td>
            </tr>
        </table>
    </div>
    </form>
</body>
```

IndexUserLogin.aspx 动态页面的“登录”按钮后台代码设计如下。

```
namespace Experiment1_学号姓名
{
    public partial class IndexUserLogin : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnLogin_Click(object sender, EventArgs e)
        {
            if (txtName.Text == "Admin" && txtPwd.Text == "123456")
            {
                lblShow.Text += "登录的用户名：" + txtName.Text + "<br/>";
                lblShow.Text += "登录的密码是：" + txtPwd.Text;
            }
            else
            {
                lblShow.Text = "登录的用户名或密码错误！";
            }
        }
    }
}
```

4. 实验分析

实验后，主要进行以下几方面的分析：比较静态页面和动态页面的页面结构的区别；静态页面和动态页面与用户交互功能的实现方法；Web的工作原理在静态页面和动态页面中的体现方法；ASP.NET 引擎的工作原理。