

Scott Mitchell



STARTER KIT

DVD includes
Microsoft Visual Web
Developer 2008
Express Edition

Sams **Teach Yourself**

ASP.NET 3.5

in **24**
Hours



SAMS

Sams Teach Yourself ASP.NET 3.5 in 24 Hours, Complete Starter Kit

Copyright © 2008 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-672-32997-5

ISBN-10: 0-672-32997-2

Library of Congress Cataloging-in-Publication Data:

Mitchell, Scott.

Sams teach yourself ASP.NET 3.5 in 24 hours : complete starter kit / Scott Mitchell.
p. cm.

ISBN 0-672-32997-2

1. Active server pages. 2. Web sites—Design. 3. Microsoft .NET. I. Title.
TK5105.8885.A26M587 2008
005.2'76—dc22

2008014770

Printed in the United States of America

First Printing June 2008

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs accompanying it.

Bulk Sales

Sams Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales

1-800-382-3419

corpsales@pearsontechgroup.com

For sales outside the U.S., please contact

International Sales

international@pearson.com

Editor-in-Chief

Karen Gettman

Executive Editor

Neil Rowe

Development Editor

Mark Renfrow

Managing Editor

Kristy Hart

Project Editor

Andrew Beaster

Copy Editor

Barbara Hacha

Indexer

Erika Millen

Proofreader

Kathy Ruiz

Technical Editor

Todd Meister

Publishing Coordinator

Cindy Teeters

Multimedia Developer

Dan Scherf

Book Designer

Gary Adair

Composition

Nonie Ratcliff



This Book Is Safari Enabled

The Safari® Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days.

Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information whenever and wherever you need it.

To gain 45-day Safari Enabled access to this book:

- ▶ Go to www.informit.com/onlineedition.
- ▶ Complete the brief registration form.
- ▶ Enter the coupon code 24DQ-UXWD-LCKF-B9DS-H2C9.

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please email customer-service@safaribooksonline.com.

Introduction

As the World Wide Web continues its meteoric growth, websites have matured from simple collections of static HTML pages to data-driven dynamic web applications. For example, websites such as eBay or Amazon.com are much more than a collection of HTML pages—they are complex applications that happen to be accessed through the Internet. Although many competing technologies exist for building data-driven websites, this book shows how to use the latest version of Microsoft's popular ASP.NET technology for creating web applications.

ASP.NET web applications are composed of individual ASP.NET web pages. As we will see in numerous examples throughout this book, these ASP.NET pages can display HTML, collect user input, and interact with databases. ASP.NET pages contain a mix of both HTML and source code. It is the source code of an ASP.NET page that allows for the more advanced features, such as accessing data from a database, or sending an email. The source code of an ASP.NET web page can be written in any one of a number of programming languages. For this book we will be using Microsoft's Visual Basic programming language. Don't worry if you've never programmed in Visual Basic, or even if you have never programmed at all. Starting with Hour 5, "Understanding Visual Basic's Variables and Operators," we spend three hours examining programming language concepts and the Visual Basic syntax.

To ease ASP.NET web page development, Microsoft provides a free development editor, Visual Web Developer, which is included in this book's accompanying CD. Visual Web Developer simplifies creating both the HTML and source code portions of ASP.NET pages. The HTML for an ASP.NET web page can be quickly created by using the Designer, which is a What You See Is What You Get (WYSIWYG) graphical editor. With the Designer, you can drag and drop various HTML elements onto an ASP.NET web page, moving them around with a few clicks of the mouse. Likewise, Visual Web Developer offers tools and shortcuts that help with creating an ASP.NET page's code.

Audience and Organization

This book is geared for developers new to ASP.NET, whether or not you've had past experience with HTML or programming languages. By the end of this book you'll be able to create and deploy your own dynamic, data-driven web applications using ASP.NET.

Sams Teach Yourself ASP.NET 3.5 in 24 Hours, Complete Starter Kit

This book's 24 hours are divided into four parts. Part I introduces you to ASP.NET, HTML, Visual Web Developer, and Visual Basic. Hour 1, "Getting Started with ASP.NET 3.5," begins with an overview of ASP.NET and then walks you through installing the .NET Framework, Visual Web Developer, and other necessary components. Hour 3, "Using Visual Web Developer," showcases Visual Web Developer, which is the powerful development editor you'll be using throughout this book to create ASP.NET web pages. Hours 5, 6, and 7 examine the syntax and semantics of the Visual Basic programming language.

ASP.NET offers a variety of user interface elements for collecting user input, including text boxes, check boxes, drop-down lists, and radio buttons. In Part II you will see how to collect and process user input. Hour 10, "Using Text Boxes to Collect Input," examines using single-line, multi-line, and password text boxes, while Hour 11, "Collecting Input Using Drop-Down Lists, Radio Buttons, and Check Boxes," examines alternative user input controls.

Part III shows how easy it is to build data-driven websites with ASP.NET. Starting in Hour 13, "An Introduction to Databases," we begin our look at building websites that interact with databases. Typically, data-driven websites enable visitors to view, update, delete, and insert data into the database from an ASP.NET page. In Hour 15, "Displaying Data with the Data Web Controls," you will learn how to display database data in a web page. Hour 16, "Deleting, Inserting, and Editing Data," examines how to edit, insert, and delete data.

Part IV highlights tools provided by ASP.NET and Visual Web Developer that help with building professional, easy-to-use websites. In Hour 20, "Defining a Site Map and Providing Site Navigation," you'll see how to define a website's navigational structure and display menus, treeviews, and breadcrumbs. Hour 22, "Using Master Pages to Provide Sitewide Page Templates," examines master pages, which enable web designers to create a web page template that can be applied to all pages across the site.

Conventions Used in This Book

This book uses several design elements and conventions to help you prioritize and reference the information it contains:

**By the
Way**

By the Way boxes provide useful sidebar information that you can read immediately or circle back to without losing the flow of the topic at hand.

Did You Know? boxes highlight information that can make your Visual Basic programming more effective.

***Did you
Know?***

Watch Out! boxes focus your attention on problems or side effects that can occur in specific situations.

***Watch
Out!***

New terms appear in a **semibold** typeface for emphasis.

In addition, this book uses various typefaces to help you distinguish code from regular English. Code is presented in a monospace font. Placeholders—words or characters that represent the real words or characters you would type in code—appear in *italic monospace*. When you are asked to type or enter text, that text appears in **bold monospace**. Menu options are separated by a comma. For example, when you should open the File menu and choose the New Project menu option, the text says “Select File, New Project.”

Some code statements presented in this book are too long to appear on a single line. In these cases, a line-continuation character ➤ is used to indicate that the following line is a continuation of the current statement. Furthermore, some code listings include line numbers. These numbers are used to refer to specific lines of code in the text and are not part of the code syntax.

I hope you enjoy reading this book as much as I enjoyed writing it.

Happy Programming!

Scott Mitchell

mitchell@4guysfromrolla.com

HOUR 1

Getting Started with ASP.NET 3.5

In this hour, we will cover

- ▶ What is ASP.NET?
- ▶ System requirements for using ASP.NET
- ▶ Software that must be installed prior to using ASP.NET
- ▶ Installing the .NET Framework, Visual Web Developer, and SQL Server 2005
- ▶ Taking a quick tour of Visual Web Developer
- ▶ Creating a simple ASP.NET web page and viewing it through a web browser

ASP.NET is an exciting web programming technology pioneered by Microsoft that allows developers to create **dynamic web pages**. Dynamic web pages are pages whose content is dynamically regenerated each time the web page is requested. For example, after you log on, the front page of Amazon.com shows books it recommends for you, based on your previous purchases. This is a dynamic web page because it is a single web page whose content is customized based on what customer is visiting. In this book we examine how to create dynamic ASP.NET websites quickly and easily.

Prior to ASP.NET, Microsoft's dynamic web programming technology was called Active Server Pages, or ASP. Although ASP was a popular choice for creating dynamic websites, it lacked important features found in other programming technologies. Microsoft remedied ASP's shortcomings with ASP.NET. ASP.NET version 1.0 was released in January 2002 and quickly became the web programming technology of choice for many. In November 2005, Microsoft released the much-anticipated version 2.0. Two years later, in November 2007, Microsoft released ASP.NET version 3.5.

Before we can start creating our first ASP.NET website, we need to install the .NET Framework, Visual Web Developer, and SQL Server 2005. The .NET Framework is a rich platform for creating Windows-based applications and is the underlying technology used to create ASP.NET websites.

Visual Web Developer is a sophisticated program for creating, editing, and testing ASP.NET websites and web pages. ASP.NET web pages are simple text files, so any text editor will suffice (such as Microsoft Notepad), but if you've created websites before, you know that using tools such as Microsoft FrontPage or Adobe Dreamweaver makes the development process much easier than using a generic text editor like Notepad. This is the case for ASP.NET, as well.

The third and final piece we'll need to install is SQL Server 2005. SQL Server is a database engine, which is a specialized application designed to efficiently store and query data. Many websites interact with databases; any e-commerce website, for example, displays product information and records purchase orders in a database. Starting with Hour 13, "An Introduction to Databases," we'll see how to create, query, and modify databases through both Visual Web Developer and ASP.NET pages.

This hour focuses on getting everything set up properly so that we can start creating ASP.NET web applications. Although it would be nice to be able to jump straight into creating ASP.NET pages, it is important that we first take the time to ensure that the pieces required for ASP.NET are correctly installed and configured. We create a very simple ASP.NET page at the end of this hour, but we won't explore it in any detail. We look at ASP.NET pages in more detail in the next hour and in Hour 4, "Designing, Creating, and Testing ASP.NET Pages."

What Is ASP.NET?

Have you ever wondered how dynamic websites like Amazon.com work behind the scenes? As a shopper at Amazon.com, you are shown a particular web page, but the web page's content is dynamic, based on your preferences and actions. For instance, if you have an account with Amazon.com, when you visit Amazon.com's home page your name is shown at the top and a list of personal recommendations is presented further down the page. When you type an author's name, a title, or a keyword into the search text box, a list of matching books appears. When you click a particular book's title, you are shown the book's details along with comments and ratings from other users. When you add the book to your shopping cart and check out, you are prompted for a credit card number, which is then billed.

Web pages in websites whose content is determined dynamically based on user input or other information are called **dynamic web pages**. Any website's search engine page is an example of a dynamic web page because the content of the search results page is based on the search criteria the user entered and the searchable documents on the web server. Another example is Amazon.com's personal recommendations. The books and products that Amazon.com suggests when you visit the home page are different from the books and products suggested for someone else. Specifically, the recommendations are determined by the products you have previously viewed and purchased.

The opposite of a dynamic web page is a **static web page**. Static web pages contain content that does not change based on who visits the page or other external factors. HTML pages, for example, are static web pages. Consider an HTML page on a website with the following markup:

```
<html>
<body>
  <b>Hello, World!</b>
</body>
</html>
```

Such a page is considered a static web page because regardless of who views the page or what external factors might exist, the output will always be the same: the text `Hello, World!` displayed in a bold font. The only time the content of a static web page changes is when someone edits and saves the page, overwriting the old version.

Virtually all websites today contain a mix of static and dynamic web pages. Rarely will you find a website that has just static web pages, because such pages are so limited in their functionality.

**By the
Way**

By learning ASP.NET, you will learn how to create websites that contain dynamic web pages. It is important to understand the differences between how a website serves static web pages versus dynamic web pages.

Competing Web Programming Technologies

ASP.NET is only one of many technologies that can be employed to generate dynamic web pages. ASP.NET is the successor to Active Server Pages (ASP), which was Microsoft's earlier dynamic web-page creation technology. Other technologies include PHP, JSP, and ColdFusion.

**By the
Way**

**By the
Way**

Personally, I find ASP.NET to be the easiest and most powerful technology of the bunch, which is why I'm writing a book about ASP.NET instead of one of the competing technologies. Moreover, the features and functionality of ASP.NET are head and shoulders above ASP. If you've created ASP pages in the past, you'll no doubt find that you can do the same things with ASP.NET but in a fraction of the time.

**Did you
Know?**

If you have experience developing web applications with other web programming technologies, such as ASP, PHP, or JSP, you may already be well versed in the material covered in the next three sections. If this is the case, feel free to skip to the "Installing the ASP.NET Engine, Editor, and Database System" section.

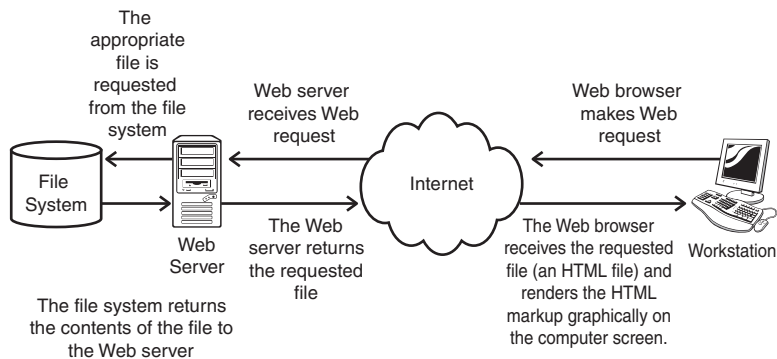
Serving Static Web Pages

If you've developed websites before, you likely know that a website requires a **web server**.

A web server is a software application that continually waits for incoming **web requests**, which are requests for a particular URL (see Figure 1.1). The web server examines the requested URL, locates the appropriate file, and then sends this file back to the client that made the web request.

FIGURE 1.1

The web server handles incoming web requests.



For example, when you visit Amazon.com, your browser makes a web request to Amazon.com's web server for a particular URL, say `/books/index.html`. Amazon.com's web server determines what file corresponds to the requested URL. It then returns the contents of this file to your browser.

This model is adequate for serving static pages, whose contents do not change. However, such a simple model is insufficient for serving dynamic pages because the

web server merely returns the contents of the requested URL to the browser that initiated the request. The contents of the requested URL are not modified in any way by the web server based on external inputs.

Serving Dynamic Web Pages

With static web pages, the contents of the web page are just HTML elements that describe how the page should be rendered in the user's web browser. Therefore, when a static web page is requested, the web server can send the web page's content, without modification, to the requesting browser.

This simple model does not work for dynamic web pages, where the content of the page may depend on various factors that can differ on a per-visitor basis. To accommodate dynamic content, dynamic web pages contain source code that is **executed** when the page is requested (see Figure 1.2). When the code is executed, it produces HTML markup as its result, which is then sent back to and displayed in the visitor's browser.

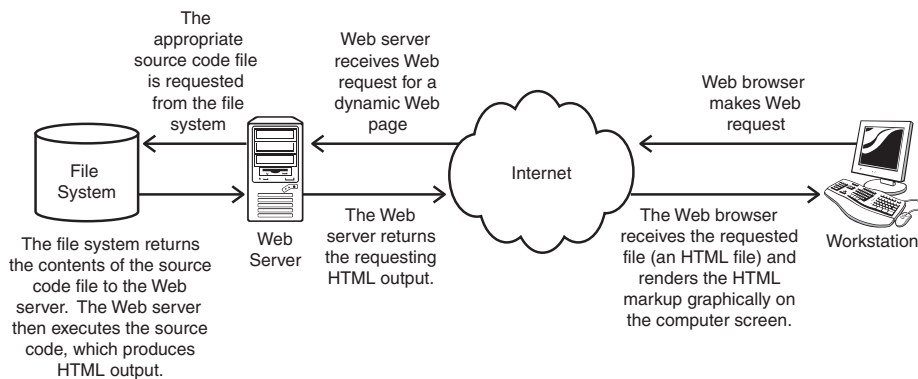


FIGURE 1.2
The content of a dynamic web page is created by executing the dynamic web page's source code.

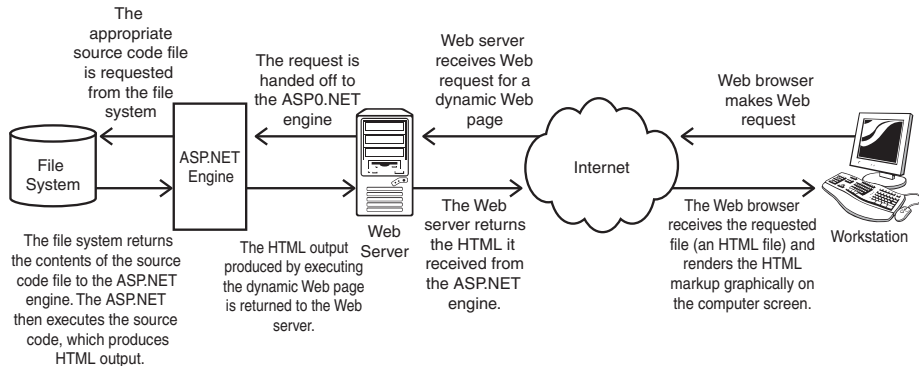
This model allows for dynamic content because the content isn't actually created until the web page is requested. Imagine that we wanted to create a web page that displays the current date and time. To do this using a static web page, someone would need to edit the web page every second, continually updating the content so that it contained the current date and time. Clearly, this isn't feasible.

With a dynamic web page, however, the executed code can retrieve and display the current date and time. Suppose that one particular user visits this page on June 12, 2008, at 4:15:03 p.m. When the web request arrives, the dynamic web page's code is executed, which obtains the current date and time and returns it to the requesting

web browser. The visitor's browser displays the date and time the web page was executed: June 12, 2008, 4:15:03 p.m. If another visitor requests this page 7 seconds later, the dynamic web page's code will again be executed, returning June 12, 2008, 4:15:10 p.m.

Figure 1.2 is, in actuality, a slightly oversimplified model. Commonly, the web server and the execution of the dynamic web page source code are decoupled. When a web request arrives, the web server determines whether the requested page is a static web page or dynamic web page. If the requested web page is static, its contents are sent directly back to the browser that initiated the request (as shown in Figure 1.1). If, however, the requested web page is dynamic—for example, an ASP.NET web page—the web server hands off the responsibility of executing the page to the **ASP.NET engine** (see Figure 1.3).

FIGURE 1.3
Execution of an ASP.NET web page is handled by the ASP.NET engine.



The web server can determine whether the requested page is a dynamic or static web page by the requested file's extension. If the extension is `.aspx`, the web server knows the requested page is an ASP.NET web page and therefore hands off the request to the ASP.NET engine.

By the Way

The ASP.NET engine is a piece of software that knows how to execute ASP.NET web pages. Other web programming technologies, such as ASP, PHP, and JSP, have their own engines, which know how to execute ASP, PHP, and JSP pages.

When the ASP.NET engine executes an ASP.NET page, the engine generates the web page's resulting HTML output. This HTML output is then returned to the web server, which then returns it to the browser that initiated the web request.

Hosting ASP.NET Web Pages

To view an ASP.NET web page that resides on a web server, we need to request it through a browser. The browser sends the request to the web server, which then dispatches the request to the ASP.NET engine. The ASP.NET engine processes the requested page and returns the resulting HTML to the browser. When you're developing ASP.NET websites, the ASP.NET web pages you create are saved on your personal computer. For you to be able to test these pages, then, your computer must have a web server installed.

Fortunately, you do not need to concern yourself with installing a web server on your computer. Visual Web Developer, the editor we'll be using throughout this book to create our ASP.NET websites, includes a lightweight web server specifically designed for testing ASP.NET pages locally. As we will see in later hours, when testing an ASP.NET page, Visual Web Developer starts the **ASP.NET Development Web Server** and launches a browser that issues a request of the form: `http://localhost:portNumber/ASP.NET_Page.aspx`.

The `http://localhost` portion of the request tells the browser to send the request to your personal computer's web server, in contrast to some other web server on the Internet. The *portNumber* specifies a particular **port** through which the request is made. All web servers listen for incoming requests on a particular port. When the ASP.NET Development Web Server is started, it chooses an open port, which is reflected in the *portNumber* portion of the URL. Finally, the `ASP.NET_Page.aspx` portion is the filename of the ASP.NET page being tested.

Hosting ASP.NET pages locally through the ASP.NET Development Web Server has a number of advantages:

- ▶ **Testing can be done while offline**—Because the request from your browser is being directed to your own personal computer, you don't need to be connected to the Internet to test your ASP.NET pages.
- ▶ **It's fast**—Local requests are, naturally, much quicker than requests that must travel over the Internet.
- ▶ **Advanced debugging features are available**—By developing locally, you can use advanced debugging techniques, such as halting the execution of an ASP.NET page and stepping through its code line-by-line.
- ▶ **It's secure**—The ASP.NET Development Web Server allows only local connections. With this lightweight web server, you don't need to worry about hackers gaining access to your system through an open website.

The main disadvantage of hosting ASP.NET pages locally is that they can be viewed only from your computer. That is, a visitor on another computer cannot enter some URL into her browser's Address bar that will take her to the ASP.NET website you've created on your local computer. If you want to create an ASP.NET website that can be visited by anyone with an Internet connection, you should consider using a web-hosting company.

Web-hosting companies have a number of Internet-accessible computers on which individuals or companies can host their websites. These computers contain web servers that are accessible from any other computer on the Internet. The benefits of using a web-hosting company to host your site include

- ▶ **A publicly available website**—With a web-hosting company, any visitor who has an Internet connection can visit your website!
- ▶ **Use of a domain name**—You can register a domain name and have it point to your website so that visitors can reach your website through a name like `www.mysite.com`.
- ▶ **Ability to focus 100% on building your website**—Installing a web server, applying the latest security patches, properly configuring domain names, and so forth can be tricky tasks. By using a web-hosting company, you are paying for this service, which enables you to concentrate on building your website.

After you have settled on a web-hosting company and have set up your account, you are ready to move the ASP.NET pages from your computer to the web-hosting company. This process is referred to as **deployment**, and is covered in-depth in Hour 24, "Deploying Your Website." After a website has been successfully deployed, you, or anyone else on the Internet, can visit the site from their web browser.

Develop Locally, Deploy to a Web Host

Because there are different advantages for hosting a site locally versus hosting with a web-hosting company, often the best choice is to do both! I encourage you to develop, test, and debug your ASP.NET websites locally, through Visual Web Developer's built-in web server. After you have completed your site and are ready to go live, you can procure an account with a web-hosting company and deploy your site. This approach allows for the best of both worlds—an ideal development environment with the end result of a publicly accessible website!

The examples and lessons presented in Hour 1 through Hour 23 are meant to be created, tested, and debugged locally. Hour 24 contains step-by-step instructions for deploying your ASP.NET website to a web-hosting company.

Installing the ASP.NET Engine, Editor, and Database System

For a web server to be able to serve ASP.NET pages, it must have the ASP.NET engine installed. Recall that the ASP.NET engine is responsible for executing the ASP.NET web page and generating its resulting HTML. To install the ASP.NET engine, your computer must be running Windows XP, Windows Server 2003, Windows Vista, or Windows Server 2008. Even if your system does have the required operating system installed, you may need to take additional steps before you can start working with ASP.NET. For example, those using Windows XP need to have Service Pack 2 (SP2) installed. If you're uncertain whether your system meets the requirements, attempt the installation process. The installation program will inform you if there is some prerequisite for installation, such as Service Pack 2.

Three components need to be installed for us to work with the ASP.NET examples throughout this book. First, we must install the .NET Framework, which contains the core libraries required to execute an ASP.NET page. The ASP.NET engine is part of this .NET Framework. Following that, we need to install Visual Web Developer, which is the editor of choice for working with ASP.NET pages. Finally, we need to install SQL Server 2005, a database engine that is used extensively from Hour 14, "Accessing Data with the Data Source Web Controls," onward.

All three components can be installed through the single installation program included on this book's accompanying CD. To begin the installation process, insert the CD into your computer. This brings up the installation program starting with the screen shown in Figure 1.4. Click the Next button to progress through the subsequent two screens.

If your computer lacks the prerequisites, such as not having the latest service pack installed, the installation program informs you of the problem and gives you instructions on how to update your system. After you have updated your computer, rerun the installation program.

**By the
Way**

The book's CD is the installation CD for Microsoft's Visual Web Developer editor. Because Visual Web Developer is designed for developing ASP.NET websites, the installation process automatically installs the .NET Framework and other required ASP.NET tools. You can also install three optional packages, as shown in Figure 1.5.

FIGURE 1.4

Start the installation process by inserting the CD into your computer's CD-ROM drive.

**FIGURE 1.5**

Make sure that you install SQL Server 2005 Express Edition.



The first optional package is MSDN Express Library for Visual Studio 2008. MSDN is Microsoft's collection of product documentation, whitepapers, code samples, and help files. Although all this information is accessible online at <http://msdn2.microsoft.com>, I encourage you to install MSDN locally on your computer. The local documentation can be searched and accessed quicker than its online counterpart and is still available even if your Internet connection is down.

The second optional package is Microsoft SQL Server 2005 Express Edition. This package is optional in the sense that Visual Web Developer will install successfully

with or without SQL Server 2005; however, the latter half of this book's examples rely on SQL Server 2005 being installed. Therefore, make sure that this check box is selected.

The third package is for Microsoft's Silverlight runtime. Silverlight is a browser plug-in from Microsoft that is capable of displaying rich, interactive multimedia content. The Microsoft Silverlight runtime is required to view Silverlight applications in your browser. None of the examples in this book use Silverlight, so you may choose to not install it.

The Microsoft SQL Server 2005 Express Edition and Microsoft Silverlight Runtime check boxes will not be displayed if you already have these components installed on your computer.

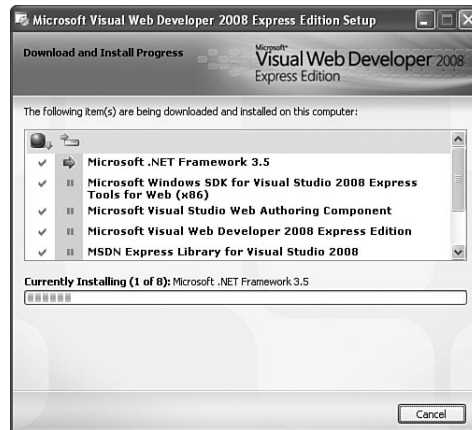
**By the
Way**

The next screen (see Figure 1.6) enables you to specify in what folder to install Visual Web Developer as well as what products will be installed and the disk space required. After double-checking that the correct packages are being installed, click the Install button to begin the installation process. The overall installation process will take several minutes. During the installation, you are kept abreast with what package is currently being installed as well as the overall installation progress (see Figure 1.7).



FIGURE 1.6
Specify the folder in which to install Visual Web Developer.

FIGURE 1.7
Monitor the
installation's
progress.



A Brief Tour of Visual Web Developer

When the installation process completes, take a moment to poke through Visual Web Developer. To launch Visual Web Developer, go to the Start menu, choose Programs, and click Microsoft Visual Web Developer 2008 Express Edition. Figure 1.8 shows Visual Web Developer when it loads.

When you open Visual Web Developer, the Start Page is initially shown. This Start Page includes a list of Recent Projects in the upper-left corner, a Getting Started section with some links for accomplishing common tasks in the bottom-left corner, and a list of recent articles on Microsoft's MSDN site in the right column.

On the left you'll find the Toolbox. When you view the Start Page, the Toolbox is empty, but when you work with an ASP.NET page, it contains the plethora of ASP.NET Web controls that can be added to the page. (We'll discuss what Web controls are and their purpose in the next hour.) Two other windows share the left region with the Toolbox: CSS Properties and Manage Styles. These windows are used to define style and appearance settings for the HTML and Web control elements within a web page.

To the right of the screen, you'll find the Solution Explorer. Again, on the Start Page this is empty, but when you load or create an ASP.NET website, the Solution Explorer will list the website's files. These files include database files, HTML pages, ASP.NET pages, image files, CSS files, configuration files, and so on. In addition to the Solution Explorer, the right portion of the screen is also home to the Database Explorer and Properties windows. The Database Explorer lists the databases associated with the project and provides functionality for creating, editing, and deleting the structure and contents of these databases. When you design an ASP.NET page, the Properties window displays information about the currently selected Web control or HTML element.

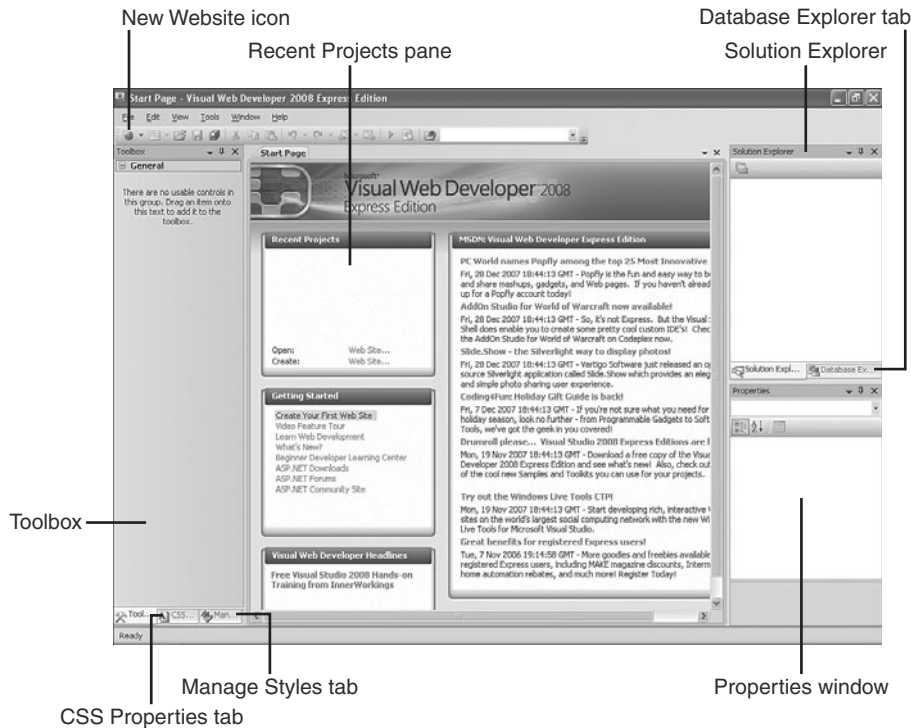


FIGURE 1.8
The Start Page shows when Visual Web Developer is loaded.

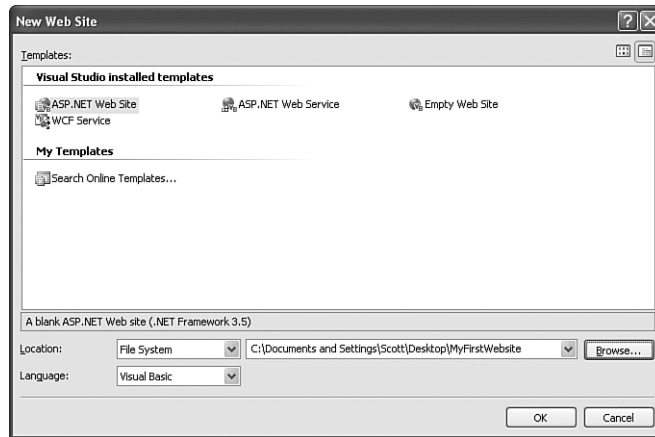
Creating a New ASP.NET Website

To create and design an ASP.NET page, we must first create an ASP.NET website. From Visual Web Developer, you can choose several ways to create a new ASP.NET website. You can go to the File menu and choose the New Web Site option; you can click the New Website icon in the Toolbar; or you can click the Create Web Site link in the Recent Projects pane of the Start Page.

All these approaches bring up the New Web Site dialog box, as shown in Figure 1.9. Let's take a moment to create a website. For now, don't worry about all the options available or what they mean because we'll discuss them in detail in Hour 3, "Using Visual Web Developer." Leave the Templates selection as ASP.NET Web Site, the Location drop-down list as File System, and the Language drop-down list as Visual Basic. The only thing you should change is the actual location of the website. Place the website in a folder named MyFirstWebsite on your desktop.

FIGURE 1.9

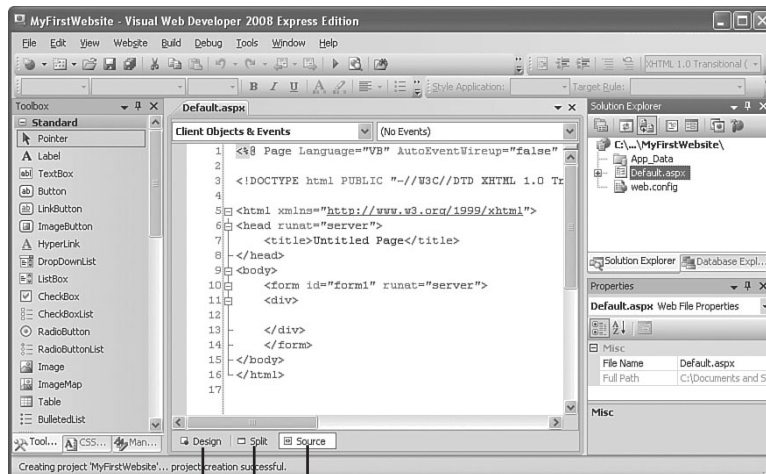
Create a new ASP.NET website in a folder on your desktop.



After you create the new website, your screen should look similar to Figure 1.10. When creating the new website, Visual Web Developer automatically created an App_Data folder, an ASP.NET page named Default.aspx, and a web configuration file (web.config). This folder and these two files are shown in the Solution Explorer.

FIGURE 1.10

A new website has been created with an ASP.NET page, Default.aspx.



Design
Split
Source

The Default.aspx page that was automatically created is opened and its contents are shown in the main window. Right now this ASP.NET page consists of just HTML. As we will see in future hours, ASP.NET pages can also contain Web controls and

server-side source code. Typically, ASP.NET pages are broken into two files: one that contains the HTML markup and Web control syntax, and another that contains just the code. In fact, if you click the plus icon on `Default.aspx` in the Solution Explorer, you'll see that there's another file, `Default.aspx.vb`, that is nested. This is the source code file for `Default.aspx`.

Don't worry if you're feeling a bit overwhelmed. The point of this hour is to give a cursory overview of Visual Web Developer. Over the next three hours, we'll look at the portions of an ASP.NET page and the steps involved in creating and testing ASP.NET pages in much greater detail.

***By the
Way***

When you're working with the HTML elements and Web controls in an ASP.NET web page, you'll notice three views. The first is the Source view, which shows the page's underlying HTML markup and Web control syntax. This is the default view, and the one shown in Figure 1.10. The second view, called the Design view, provides a simpler alternative to specifying and viewing the page's content. In the Design view you can drag and drop HTML elements and Web controls from the Toolbox onto the design surface. You don't need to type in the specific HTML or Web control syntax. The third view, Split, divides the screen in half, showing the Source view in the top portion and the corresponding Design view in the bottom. You can toggle between the Source, Design, and Split views for an ASP.NET page by using the Design, Source, and Split buttons at the bottom of the main window.

You can reposition the Properties, Toolbox, Solution Explorer, Database Explorer, and other windows by clicking their title bars and dragging them elsewhere, or resize them by clicking their borders. If you accidentally close one of these windows by clicking the X in the title bar, you can redisplay it from the View menu.

***Did you
Know?***

Creating and Testing a Simple ASP.NET Web Page

To view or test an ASP.NET web page, a browser needs to make a request to the web server for that web page. Let's test `Default.aspx`. Before we do, though, let's add some content to the page, because right now the page's HTML will not display anything when viewed through a browser. From the Source view, place your cursor between the `<div>` and `</div>` tags in `Default.aspx` and add the following text:

```
<h1>Hello, World!</h1>
```

This displays the text **Hello, World!** in a large font. After entering this text, go to the Debug menu and choose the Start Without Debugging menu option. This starts the

ASP.NET Development Web Server and launches your computer's default browser, directing it to `http://localhost:portNumber/MyFirstWebsite/Default.aspx` (see Figure 1.11). The *portNumber* portion in the URL will depend on the port selected by the ASP.NET Development Web Server.

FIGURE 1.11
Default.aspx, when viewed through a browser.



This ASP.NET page isn't very interesting because its content is static. It does, however, illustrate that to view the contents of an ASP.NET page, you must start the ASP.NET Development Web Server and request the page through a browser.

By the Way

You may be wondering what debugging is, and why I instructed you to start without it. We'll cover the differences between the Start Debugging and Start Without Debugging menu options in Hour 4.

Summary

Today virtually all websites contain dynamic web pages of one kind or another. Any website that allows a user to search its content, order products, or customize the site's content is dynamic in nature. A number of competing technologies exist for creating dynamic pages—one of the best is ASP.NET.

Throughout this book we'll be examining how to create interactive and interesting ASP.NET web pages. You will implement the examples using Visual Web Developer, a free editor from Microsoft designed specifically for working on ASP.NET websites. In the "Installing the ASP.NET Engine, Editor, and Database System" section, we looked at how to install Visual Web Developer, along with the .NET Framework and SQL Server 2005. In the "A Brief Tour of Visual Web Developer" section, we poked around Visual Web Developer and created our first ASP.NET web page, testing it

through a browser. In Hour 3 we'll explore the Visual Web Developer environment in much greater detail.

We have just begun our foray into the world of ASP.NET. Over the next 23 hours we'll explore the ins and outs of this exciting technology!

Q&A

Q. *What is the main difference between a static and a dynamic web page?*

- A.** A static web page has content that remains unchanged between web requests, whereas a dynamic web page's content is generated each time the page is requested. The dynamic content is typically generated from user input, data-base data, or some combination of the two.

For an example of a dynamic web page, consider the official website for the National Basketball Association (www.NBA.com), which lists team schedules, ongoing game scores, player statistics, and so on. This site's web pages display information stored in a database. Another example is a search engine site such as Google, which displays dynamic content based on both its catalog of indexed websites and the visitor's search terms.

Q. *You mentioned that with Visual Web Developer we'll be using the ASP.NET Development Web Server. Are other web server systems available for serving ASP.NET pages?*

- A.** The ASP.NET Development Web Server is designed specifically for testing ASP.NET web pages locally. The computers at web-hosting companies use different web server software. Many web-hosting companies use Microsoft's Internet Information Server (IIS), which is a professional-grade web server designed to work with Microsoft's dynamic web technologies—ASP and ASP.NET.

If you are running Windows XP Professional, Windows Server 2003, Windows Vista, or Windows Server 2008, IIS may already be installed on your computer. If not, you can install it by going to Start, Settings, Control Panel, Add or Remove Programs, and clicking the Add/Remove Windows Components option.

I encourage you not to install IIS and to instead use the ASP.NET Development Web Server unless you are already familiar with IIS and know how to administer and secure it. The ASP.NET Development Web Server is more secure because it allows only incoming web requests from the local computer; IIS, on the other hand, is a full-blown web server and, unless properly patched and administered, can be an attack vector for malicious hackers.

Workshop

Quiz

1. What is the difference between a static web page and a dynamic web page?
2. What is the purpose of the ASP.NET engine?
3. True or False: ASP.NET web pages can be served from computers using the Windows ME operating system.
4. What software packages must be installed to serve ASP.NET web pages from a computer?
5. When should you consider using a web-hosting company to host your ASP.NET web pages?

Answers

1. The HTML markup for a static web page remains constant until a developer actually modifies the page's HTML. The HTML for a dynamic web page, on the other hand, is produced every time the web page is requested.
2. When the web server receives a request for an ASP.NET web page, it hands it off to the ASP.NET engine, which then executes the requested ASP.NET page and returns the rendered HTML to the web server. The ASP.NET engine allows for the HTML of an ASP.NET web page to be dynamically generated for each request.
3. False. ASP.NET pages can be served only from computers running Windows XP, Windows Server 2003, or Windows Vista.
4. For a computer to serve ASP.NET web pages, the .NET Framework and a web server that supports ASP.NET must be installed.
5. You should consider using a web-hosting company if you want your ASP.NET website to be accessible via the Internet. Typically, it's best to first develop the website locally and then deploy it to a web-hosting company when you are ready to go live.

Exercises

This hour does not have any exercises. We'll start with exercises in future hours, after we become more fluent with creating ASP.NET web pages.

Index

Symbols

+ (addition) operator, 117
' (apostrophe), 98
&= assignment operator, 122
+= assignment operator, 122
*= assignment operator, 122
-= assignment operator, 122
/= assignment operator, 122
= assignment operator, 110-111, 120-121
* (asterisk), 328
& (concatenation) operator, 119-120
/ (division) operator, 117
= (equal) operator, 119
> (greater than) operator, 119
>= (greater than or equal) operator, 119
< (less than) operator, 119
<= (less than or equal) operator, 119
* (multiplication) operator, 117
<> (not equal) operator, 119
- (subtraction) operator, 117
4GuysFromRolla.com, 74

A

Access, 299
access modifiers, 154
access rules (users), 517-519
AccessDataSource control, 321
AccessingData.aspx web page, 321, 337
accounts. See user accounts
Active Server Pages (ASP), 7
Add Item dialog box, 486
Add New Item dialog box, 65-66
Add WHERE Clause dialog box, 339
addition (+) operator, 117
AJAX
 advantages of, 571
 ASP.NET AJAX Control Toolkit, 577
 browser compatibility, 577
 definition, 564
 examples, 566
 exercises, 579
 loading images, 576
 overview, 563-566
 partial page
 postback, 564-565

AJAX

ScriptManager control, 567

UpdatePanel control

- adding to web pages, 568-570
- definition, 567
- multiple controls, 571-573
- overview, 567
- progress messages, displaying, 574-576
- refreshing, 573

UpdateProgress control, 567

AJAXSimple.aspx web page, 568-570

AllowPaging property

- DetailsView control, 362
- GridView control, 365

AllowPaging property (FormView control), 471

AlternatingItemTemplate (ListView control), 459

AlternatingRowStyle property (GridView control), 356

Amazon.com website, 8

ampersand (&), 119-120

anonymous users, 518

Answer property (CreateUserWizard), 523

AnswerLabelText property (CreateUserWizard), 523

apostrophe ('), 98

App_Data folder, 31, 304

arithmetic operators, 117-118

ASP (Active Server Pages), 7

ASP.NET AJAX Control Toolkit, 577

ASP.NET AJAX framework. See **AJAX**

ASP.NET Developer Center, 74

ASP.NET Development Web Server, 13-14

ASP.NET engine

- definition, 12
- installation, 15-18

ASP.NET resources, 74

ASP.NET version 1.0, 7

ASP.NET version 2.0, 7

ASP.NET web pages. See **web pages**

ASP.NET Website Administration Tool, 510, 600

- access rules, 517-519
- authentication settings, 511-514
- classifying users by role, 516-517
- creating users, 514
- managing users, 515
- SMTP settings, 520-521

ASP.NET websites. See **websites**

ASPNETDB database, 512

ASPNETDB.MDF database, 513

.aspx file extension, 12

assigning values to variables, 110-111, 116

assignment operators, 110-111, 120-121, 127

AssociatedUpdatePanelID property (UpdateProgress control), 575

asterisk (*), 328

authentication

- displaying content based on authentication status, 532-533
- forms-based authentication, 511
- Windows authentication, 511

Auto-increment columns, 303

AutoFormat dialog box, 359-360, 494-495

AutoGenerateColumns property (GridView control), 353

automatic postback (list Web controls), 415

AutoPostBack property (list Web controls), 415

AutoRecover feature (Visual Web Developer), 69

B

**** tag (HTML), 26

BackColor property

- DropDownList control, 246
- Font control, 246
- GridView control, 354
- Label control, 180-182
- TextBox control, 228

BackColorUrl property (GridView control), 354

binary operators, 117

binding. See **data binding**

bit columns, 431-434

blank strings, 392

BMI (Body Mass Index) calculator

- BMICalculator.aspx, 202-203
- processing, 206-208
- testing, 204-206
- writing code for, 208-210

form element, 196-199

input element, 194-196

body of subroutines, 146

Body Mass Index calculator. See **BMI calculator**

bold, HTML markup for, 26

Bold subproperty (Label control Font property), 178

Books table, 312-315

BooksDataSource control, 454

Boolean data type, 115

BorderColor property

GridView control, 354

Label control, 183

TextBox control, 228

BorderStyle property

GridView control, 354

Label control, 182-183

TextBox control, 228

BorderWidth property

GridView control, 354

Label control, 184

SiteMapPath control, 493

TextBox control, 228

BoundField, 430

**
 tag (HTML), 27**

breadcrumbs, 484, 491-492

breakpoints

definition, 100

setting, 100-101

browsers, AJAX compatibility, 577

ButtonField, 431

ByRef keyword, 149

ByVal keyword, 149

C

calling

functions, 155

methods, 164-165

subroutines, 146

cascading style sheets (CSS), 35

case sensitivity, 54

casting data types

explicit casting, 123-124

implicit casting, 123-126

narrowing casts, 124

widening casts, 124

CausesValidation property, 291

ChangePassword control, 535

char data type, 302

check boxes

adding to web

pages, 252-253

default selections, 258

definition, 252

determining which

check boxes are

selected, 254-255

exercises, 259

CheckBox Web control

adding to web

pages, 252-253

Checked property,

254-255, 258

default selections, 258

definition, 252

determining which

check boxes are

selected, 254-255

exercises, 259

CheckBox.aspx page, 252-253

CheckBoxField, 430, 433-434

CheckBoxList Web control

customizing appearance

of, 424-425

enumerating list items, 422

overview, 421-422

SelectedItem and

SelectedValue

properties, 423-424

Checked property (CheckBox control), 254-255, 258

Choose Location dialog box, 61-62

choosing

Web controls, 237-238

web-hosting

companies, 582-583

classes

constructors

creating objects with, 162

default constructors, 163

definition, 162

naming conventions, 162

parameters, 163

creating, 168

definition, 46, 160

MailMessage, 168

methods

calling, 164-165

definition, 162

ExecuteReader, 165

overloaded methods, 163

SmtpClient, 168

SqlCommand, 163-165

classifying users by role, 516-517

clauses. See keywords

client-side scripts, 64, 272

client-side validation, 272

closing Visual Web Developer windows, 71

closing tags (HTML), 26

code redundancy, 146-147

coercion. See implicit casting

colors

HTML colors, 181

Label property, 180-182

in text boxes, 228-229

Visual Web Developer, 70

columns

adding/removing, 316

Auto-increment columns, 303

bit columns, 431-434

columns

- column types, 301-302
- definition, 300
- primary key
 - columns, 302-303
- Columns property (TextBox control), 225-226**
- CommandField, 431**
- commands**
 - Debug menu
 - Continue, 102
 - Start Debugging, 99
 - Start Without Debugging, 21
 - Step Over, 102
 - Stop Debugging, 102
 - File menu
 - New Web Site, 19, 60
 - Open Web Site, 63
 - Recent Projects, 63
 - Table menu, Insert Table, 547
- commercial database systems, 298-299**
- CompareValidator control**
 - adding to web pages, 275
 - comparing input with, 280-282
 - ControlToValidate property, 277
 - ErrorMessage property, 277
 - Operator property, 276-277
 - testing, 278-279
 - Type property, 277
 - ValueToCompare property, 277
- comparing input with CompareValidator control, 280-282**
- comparison operators, 118-119, 334**
- comparison validation, 263**
- Compute Monthly Cost button (financial calculator), 83-84**
- ComputeCostPerMonth function, 151-152**
- concatenation operator, 119-120**
- conditional control structures. See If statement**
- Configure Data Source Wizard, 322-325**
- configuring websites**
 - for membership support, 510-514
 - STMP settings, 519-521
- confirmation messages, displaying, 401**
- ConfirmPasswordLabelText property (CreateUserWizard), 523**
- connection strings**
 - definition, 322
 - updating in web.config files, 596-598
- ConnectionString property (SqlDataSource control), 327**
- constructors**
 - creating objects with, 162
 - default constructors, 163
 - definition, 162
 - naming conventions, 162
 - parameters, 163
- content pages**
 - creating, 550-552
 - passing information between content and master pages, 560
- ContentPlaceHolder control, 544**
- Continue command (Debug menu), 102**
- ContinueButtonText property (CreateUserWizard), 524**
- control structures**
 - definition, 131
 - functions
 - calling, 155
 - compared to subroutines, 154
 - ComputeCostPerMonth, 151-152
 - definition, 142
 - parameters, 148
 - syntax, 150
 - If statement
 - Else clause, 136-137
 - Elseif clause, 137-138
 - example, 133-135
 - syntax, 133
 - loops
 - Do ... Loop, 141-142
 - For ... Next, 139-141
 - infinite loops, 142
 - overview, 139
 - overview, 132
 - subroutines
 - body, 146
 - calling, 146
 - compared to functions, 154
 - creating, 146
 - definition, 142
 - example, 142-145
 - parameters, 148-150
 - ShowWelcomeMessage, 146-150
- controls. See Web controls**
- ControlToValidate property (RequiredFieldValidator control), 268-269**
- copying drop down lists, 257**
- CreateAccount.aspx web page, 521-522**

CreateUserButtonText property
(CreateUserWizard), 524

CreateUserUrl property (Login control), 530

CreateUserWizard Web control, 521-522

customizing, 522-524
prompting for email address
or password, 524-525

credentials, 510

CSS (cascading style sheets), 35

CssClass property

GridView control, 354
SiteMapPath control, 493

current nodes (site maps), 493

CurrentNodeStyle property
(SiteMapPath control), 494

customizing

CheckBoxList
control, 424-425

CreateUserWizard
control, 522-524

DetailsView control, 363,
369, 399

FormView templates,
473-474

GridView

AutoFormat dialog
box, 359-360

fields, 356-359

overview, 353-354

Properties window,
354-356

GridView control, 369

Menu control, 503-504

RadioButtonList control,
424-425

SiteMapPath control,
492-494

TreeView control, 497-500

user account features, 535
Visual Web Developer, 68-71

CustomValidator control, 289

D

**Dashed value (BorderStyle prop-
erty)**, 183

data binding

definition, 443

list Web controls, 409-410

one-way data binding,
443-444

specifying for Web
controls, 444-449

two-way data
binding, 443-444

data source controls

AccessDataSource
control, 321

overview, 319-321

SqlDataSource control.
See also SQL

adding to web
pages, 321-322

configuring data
sources, 322-325

ConnectionString
property, 327

filtering data, 337-340

ID property, 326

markup, 342

SelectCommand
property, 327

testing queries, 340-341

data types, 112-113

Boolean, 115

casting

explicit casting, 123-124

implicit casting, 123-126

narrowing casts, 124

widening casts, 124

char, 302

DateTime, 115

definition, 110

Double, 114

Integer, 113

Long, 113

loose typing, 112

nchar, 302

nvarchar, 302

Object, 115-116

Short, 114

Single, 114

String, 115

strings, 302

strong typing, 113

validation, 263

varchar, 302

data Web controls

definition, 347

DetailsView control

customizing, 363, 369

exercises, 371

overview, 360-361

paging interface, 362

GridView control

customizing,
353-360, 369

declarative markup,
352-353

exercises, 371

online resources, 370

overview, 351-352

paging interface, 364-366

sorting data, 366-368

overview, 347-351

**Database Design for Mere
Mortals**, 311

Database Publishing Wizard

Database Publishing Wizard, 592-594

databases

- accessing with data source controls

- AccessDataSource control, 321

- overview, 319-321

- SqlDataSource control, 321-327, 337-342

- ASPNETDB, 512

- ASPNETDB.MDF, 513

- commercial database systems, 298-299

- creating, 304-306

- definition, 297

- deleting data

- with DELETE statement, 379

- with GridView control, 381-384

- with SqlDataSource control, 374-377

- design, 308-311

- columns, 309-310

- logical entities, 310-311

- exercises, 318

- inserting data

- with DetailsView control, 398-400

- with INSERT statement, 378-379

- with SqlDataSource control, 374-377

- overview, 298

- populating drop-down lists from, 257

- replicating, 591-592

- with Database Publishing Wizard, 592-594

- with output scripts, 594-596

- retrieving data from. See SELECT statement

- SQL Server, 8

- tables

- adding data to, 312-315

- columns, 300-303, 316

- creating, 306-308

- definition, 300

- records, 300

- updating data

- with GridView control, 387-397

- with SqlDataSource control, 374-377

- with UPDATE statement, 379-380

- DataFormatString property (GridView control), 358**

- DataKeyNames property (GridView control), 353, 387**

- DataSourceID property (GridView control), 353**

- DateTime data type, 115**

- DB2, 299**

- Debug menu commands**

- Continue, 102

- Start Debugging, 99

- Start Without Debugging, 21

- Step Over, 102

- Stop Debugging, 102

- testing web pages, 39-40

- debugger**

- breakpoints

- definition, 100

- setting, 100-101

- definition, 91, 99

- Locals window, 102

- running, 99-102

- stopping, 102

- Watch window, 102

- Debugging Not Enabled dialog box, 99-100**

- declaring variables, 111, 116-117**

- decrementing loops, 140**

- default constructors, 163**

- default events, 103**

- Default.aspx page, 20, 31**

- Default.aspx.vb file, 31**

- DELETE statement, 379**

- deleting data**

- with DELETE statement, 379

- with GridView

- control, 381-384

- with SqlDataSource control, 374-377

- table columns, 316

- website files, 67

- deploying websites**

- building sample Web applications, 585-589

- definition, 14

- deployment process, 584-585

- hosting websites from personal computers, 584

- overview, 581

- replicating databases, 591-592

- with Database Publishing Wizard, 592-594

- with output scripts, 594-596

- updating web.config settings, 596-598

- uploading website files, 590-591

- visiting remote website, 598-599

- web-hosting companies, 582-583

- DESC clause (SELECT statement), 337**

designing web pages

- database design, 308-311
 - columns, 309-310
 - logical entities, 310-311
- design requirements
 - features, 78-79
 - overview, 77-78
 - user interfaces, 79-80

Design view (Visual Web Developer), 30, 33-40**DestinationPageUrl property (Login control), 530****DetailsView control**

- customizing, 363, 369
- exercises, 371
- fields
 - BoundField, 430
 - ButtonField, 431
 - CheckBoxField, 430, 433-434
 - CommandField, 431
 - HyperLinkField, 431, 434-437
 - ImageField, 431, 438-439
 - overview, 430
 - TemplateField, 431, 451
- inserting data with, 398-400
- overview, 360-361
- paging interface, 362

DetailsView.aspx web page, 360**dialog boxes**

- Add Item, 486
- Add New Item, 65-66
- Add WHERE Clause, 339
- AutoFormat, 359-360, 494-495
- Choose Location, 61-62
- Debugging Not Enabled, 99-100
- Insert Table, 547
- ListItem Collection Editor, 241

Modify Style, 548

- New Web Site, 19-20, 31, 60
- Open Web Site, 63
- Options (Visual Web Developer), 68-70
- Select a Master Page, 550
- Style Builder, 548

Dim statement, 111, 116-117**directives, @Page, 32****Display property**

- validation controls, 288
- validation controls, 286-291

DisplayAfter property

(UpdateProgress control), 575

displaying data. See data**Web controls****DisplayRememberMe property (Login control), 530****"Dissecting the Validation Controls" (article), 291****division operator, 117****Do ... Loop loops, 141-142****Dotted value (BorderStyle property), 183****Double data type, 114****Double value (BorderStyle property), 183****downloading Microsoft SQL Server Management Studio Express Edition, 594-595****drop-down lists**

- adding list items to, 238-239
- adding to web pages, 239-245
- copying, 257
- overview, 238
- populating from databases, 257
- properties, 246
- rearranging order of items, 243

DropDownList Web control

- adding list items to, 238-239
- adding to web pages, 239-245
- BackColor property, 246
- copying, 257
- filtering data with, 418-420
- Font property, 246
- ForeColor property, 246
- Items property, 241
- listing genres in, 417-418
- overview, 238, 416
- populating from databases, 257
- rearranging order of items, 243
- Show All Genres option, 427

DropDownList.aspx page, 239, 244-245**Dynamic Help (Visual Web Developer), 72-73****dynamic web pages. See also web pages**

- compared to static web pages, 23
- definition, 7-9, 64
- serving, 11-12

DynamicEnableDefaultPopOutImage property (Menu control), 503**DynamicHorizontalOffset property (Menu control), 504****DynamicItemFormatString property (Menu control), 504****DynamicPopOutImageTextFormatString property (Menu control), 504****DynamicPopOutImageUrl property (Menu control), 504****DynamicVerticalOffset property (Menu control), 504**

E

editable GridViews,
creating, 385-387

editing

- HTML with Visual Web Developer Design view, 33-40
- with SqlDataSource control, 374-377
- with UPDATE statement, 379-380

EditItemTemplate

- FormView control, 472
- ListView control, 459

EditRowStyle property (GridView control), 356

elements

- , 26
-
, 27
- definition, 26
- end tags, 26
- <form>, 196-199
 - example, 198
 - form submission, 197
 - postback forms, 199-200
 - redirect forms, 199-200
- <hr>, 27
- input, 194-196
- nested tags, 29
- <siteMapNode>, 487-490
- start tags, 26
- <table>
 - adding to pages, 33-40
 - definition, 33
 - when to use, 55

Else clause (If statement), 136-137

Elseif clause (If statement), 137-138

email

- sending from ASP.NET pages, 168
- sending to new user accounts, 525-527

email addresses, prompting users to enter, 524-525

Email property (CreateUserWizard), 523

EmailLabelText property (CreateUserWizard), 523

EmailRequiredErrorMessage property (CreateUserWizard), 524

empty strings, 143

EmptyDataRowStyle property (GridView control), 356

EmptyDataTemplate

- FormView control, 472
- ListView control, 459

EmptyDataText property (GridView control), 354

Enabled property (drop-down list items), 242

end tags (HTML), 26

equal operator, 119

erroneous input, testing, 97-98

error messages, displaying with RequiredFieldValidator control, 270

event-driven programming

- event handlers, 47-51
- out-of-order execution, 48-49
- raising events, 48

event handlers, 47-48, 153

- creating, 50-51, 166
- PerformCalcButton_Click, 87-88
- SelectedIndexChanged, 412-415

events, 165

- default events, 103
- definition, 47
- event-driven programming
 - event handlers, 47-51
 - out-of-order execution, 48-49
 - raising events, 48
- event handlers, 47-48, 153
 - creating, 50-51, 166
 - PerformCalcButton_Click, 87-88
 - SelectedIndexChanged, 412-415
- raising, 48
- RowDeleted, 384
- RowDeleting, 384

“Examining ASP.NET’s Membership, Roles, and Profile” (article), 535

“Examining ASP.NET’s Site Navigation” (article), 506

ExecuteReader method, 165

executing

- code, 11
- event handlers, 48

ExpandImageTooltip property (TreeView control), 499

explicit casting, 123-124

expressions, regular, 284

eXtensible Markup Language. See XML files, 484

F**fields (control)**

- DetailsView Web control
 - BoundField, 430
 - ButtonField, 431

- CheckBoxField, 430, 433-434
- CommandField, 431
- HyperLinkField, 431, 434-437
- ImageField, 431, 438-439
 - overview, 430
- TemplateField, 431, 451
- GridView control, 356-359
- GridView Web control
 - BoundField, 430
 - ButtonField, 431
 - CheckBoxField, 430, 433-434
 - CommandField, 431
 - HyperLinkField, 431, 434-437
 - ImageField, 431, 438-439
 - overview, 430
 - TemplateField, 431, 451
- fields (database). See columns**
- File menu commands**
 - New Web Site, 19, 60
 - Open Web Site, 63
 - Recent Projects, 63
- files**
 - adding to websites, 64-67
 - Default.aspx, 31
 - Default.aspx.vb, 31
 - deleting, 67
 - image files, 64
 - moving, 67
 - renaming, 67
 - script files, 64
 - style sheet files, 64
 - uploading, 590-591
 - Web.config, 31, 64, 596-598
 - Web.sitemap, 486
- XML files
 - definition, 320, 484
 - siteMapNode element, 487-490
- filtering data**
 - with DropDownList control, 418-420
 - with RadioButtonList Web control, 426
 - with SqlDataSource control, 337-339
- financial calculator web page (FinancialCalculator.aspx)**
 - Compute Monthly Cost button, 83-84
 - design requirements
 - features, 78-79
 - user interface, 79-80
 - Label Web control, 84-86
 - overview, 80-81
 - PerformCalcButton_Click event handler, 87-88
 - source code, 98-99
 - code listing, 89-91
 - viewing, 92-97
 - writing, 87-88
 - testing, 91-92
 - testing erroneous input, 97-98
 - viewing rendered source code, 92-97
 - TextBox Web controls
 - inserting, 81-83
 - reading values in, 88-89
- FinancialCalculator.aspx. See financial calculator web page**
- firing events, 48**
- folders**
 - adding to websites, 64-67
 - App_Data, 31, 304
- Font property**
 - GridView control, 354
 - Label control, 178-179, 184-185
 - SiteMapPath control, 493
 - TextBox control, 228
- fonts**
 - Label property, 178-179, 184-185
 - in text boxes, 228-229
 - Visual Web Developer, 70
- footer region (master pages), 550**
- FooterStyle property (GridView control), 356**
- FooterTemplate (FormView control), 472**
- For ... Next loops, 139-141**
- ForeColor property**
 - Font control, 246
 - GridView control, 355
 - Label control, 180-182
 - SiteMapPath control, 493
 - TextBox control, 228
- <form> element, 196-199**
 - example, 198
 - form submission, 197
 - postback forms, 199-200
 - redirect forms, 199-200
- forms. See Web Forms**
- forms-based authentication, 511**
- FormView control**
 - available templates, 472
 - custom templates, 473-474
 - exercises, 479
 - overview, 469-470
 - paging records, 470-471
- functions. See also methods; procedures**
 - calling, 155
 - compared to subroutines, 154

functions

ComputeCostPerMonth,
151-152
definition, 142
parameters, 148
syntax, 150

G

general nodes (site maps), 493

Gmail, 566

Google

Gmail, 566
Suggest, 566

greater than operator, 119

greater than or equal
operator, 119

GridLines property (GridView
control), 355

GridView control

customizing, 369
 AutoFormat dialog
 box, 359-360
 fields, 356-359
 overview, 353-354
 Properties window,
 354-356
declarative markup, 352-353
deleting data with, 381-384
editable GridViews, 385-387
editing data with, 387-397
 custom editing interface,
 392-397
 editing and formatting
 fields, 391-392
 null strings/blank
 strings, 392
 read-only fields, marking,
 389-390
exercises, 371, 403-405

online resources, 370
overview, 351-352
paging interface, 364-366
sorting data, 366-368

“GridView Examples for ASP.NET”
(article), 370

GridView Web control fields

BoundField, 430
ButtonField, 431
CheckBoxField, 430, 433-434
CommandField, 431
HyperLinkField, 431, 434-437
ImageField, 431, 438-439
overview, 430
TemplateField, 431, 451

Groove value (BorderStyle
property), 183

grouping radio buttons, 249-250

GroupName property
(RadioButton control), 249-250

GroupTemplate (ListView control),
459, 476

H

Handles keyword, 166

Head First SQL, 311

headers (master pages), 547-548

HeaderStyle property (GridView
control), 356-358

HeaderTemplate (FormView
control), 472

HeaderText property (GridView
control), 358

Height property (Label
control), 185

help

MSDN Library, 72
Visual Web Developer, 72-73

HelpPageUrl property (Login
control), 530

horizontal lines, HTML
markup, 27

hosting web pages, 61

through ASP.NET Development
Web Server, 13-14
from personal
computers, 584
through web-hosting
companies, 14, 74,
582-583

HoverNodeStyle property
(TreeView control), 498

“How To Use Membership in
ASP.NET” (article), 535

<hr> tag (HTML), 27

HTML (Hypertext Markup
Language)

colors, 181
definition, 26
editing with Visual Web
Developer Design
view, 33-40
elements
 , 26

, 27
 definition, 26
 end tags, 26
 <form>, 196-200
 <hr>, 27
 input, 194-196
 nested tags, 29
 <siteMapNode>, 487-490
 start tags, 26
 <table>, 33-40
 when to use, 55
overview, 26
viewing in Visual Web
Developer Source
view, 40-45
whitespace, 27-29

HtmlEncode property (GridView control), 358

HyperLinkField, 431, 434-437

hyperlinks, displaying with
HyperLinkField, 434-437

Hypertext Markup Language.
See HTML

I

IBM DB2, 299

IBM Informix, 299

ID property

Literal control, 173

SqlDataSource control, 326

If statement

Else clause, 136-137

Elseif clause, 137-138

example, 133-135

syntax, 133

IIS (Internet Information
Server), 23

image files, 64

Image Web controls (Visual Web
Developer), 37

ImageField, 431, 438-439

images, displaying with
ImageField, 438-439

ImageSet property (TreeView con-
trol), 499

ImageUrl property (Image web
control), 38

immutable values, 110

implicit casting, 123-126

improperly nested tags
(HTML), 29

inactive user accounts, 527-528

incrementing loops, 140

inequality operator, 119

infinite loops, 142

Informix, 299

inheriting from master pages,
552-553

input element, 194-196

input. See user input

INSERT statement, 378-379

Insert Table command (Table
menu), 547

Insert Table dialog box, 547

inserting data

with DetailsView control,
398-400

with INSERT statement,
378-379

with SqlDataSource control,
374-377

InsertItemTemplate

FormView control, 472

ListView control, 459

Inset value (BorderStyle
property), 183

installation

ASP.NET engine, 15-18

.NET Framework, 15-18

Visual Web Developer, 15-18

instantiation, 88

instructions, 108

Integer data type, 113

integers, 113-114

IntelliSense, 53

Internet Information
Server (IIS), 23

intranets, 511

Italic subproperty (Label control
Font property), 178

Items property (DropDownList
control), 241

ItemSeparatorTemplate (ListView
control), 459

ItemStyle property (GridView
control), 358

ItemTemplate

FormView control, 472

ListView control, 455-459

ItemWrap property (Menu
control), 504

J-K

JavaScript, 566

keywords

ByRef, 149

ByVal, 149

DESC, 337

Else, 136-137

Elsel, 137-138

Handles, 166

ORDER BY, 335-336

Private, 154

Protected, 155

Step, 140

WHERE, 333-335,
440-442, 450

L

Label Web control

BackColor property, 180-182

BorderColor property, 183

borders, 182-184

BorderStyle property,
182-183

BorderWidth property, 184

colors, 180-182

exercises, 190

financial calculator web
page, 84-86

Label Web control

- Font property, 178-179, 184-185
- fonts, 178-179, 184-185
- ForeColor property, 180-182
- Height property, 185
- setting properties
 - programmatically, 188
- Text property, 177
- ToolTip property, 185
- tooltips, 185
- Visible property, 186
- Width property, 185
- labels. See Label Web control**
- languages, 62-63, 108-109**
- LayoutTemplate (ListView control), 455-459**
- leaf nodes (TreeView), 497**
- LeafNodeStyle property (TreeView control), 498**
- length of text boxes, 225-226**
- less than operator, 119**
- less than or equal operator, 119**
- LevelStyles property (TreeView control), 498**
- libraries, MSDN Library, 16, 72**
- line breaks, HTML markup, 27**
- list items**
 - adding to DropDownList control, 238-239
 - definition, 238
 - rearranging order of, 243
- list Web controls**
 - automatic postback, 415
 - binding data to, 409-410
 - CheckBoxList
 - customizing appearance of, 424-425
 - enumerating list items, 422
 - overview, 421-422
 - SelectedItem and SelectedValue properties, 423-424
 - definition, 408
 - DropDownList
 - adding list items to, 238-239
 - adding to web pages, 239-245
 - copying, 257
 - filtering data with, 418-420
 - listing genres in, 417-418
 - overview, 238, 416
 - populating from databases, 257
 - properties, 246
 - rearranging order of items, 243
 - Show All Genres option, 427
 - exercises, 428
 - ListView control
 - adding templates manually, 460-462
 - adding templates with Configure ListView option, 458-460
 - AlternatingItemTemplate, 459
 - declarative markup, 456
 - EditItemTemplate, 459
 - EmptyDataTemplate, 459
 - exercises, 478
 - GroupTemplate, 459, 476
 - InsertItemTemplate, 459
 - ItemSeparatorTemplate, 459
 - ItemTemplate, 455-459
 - LayoutTemplate, 455-459
 - overview, 454-455
 - paging support, 465-467
 - sorting interface, 464-465
 - when to use, 476
- overview, 408-409
- populating, 411-412
- programmatically responding to changed selections, 412-415
- RadioButtonList**
 - customizing appearance of, 424-425
 - enumerating list items, 422
 - overview, 421-422
 - SelectedItem and SelectedValue properties, 423-424
- ListControls.aspx web page, 409-410**
- ListItem Collection Editor dialog box, 241**
- ListView control**
 - adding templates manually, 460-462
 - adding templates with Configure ListView option, 458-460
 - AlternatingItemTemplate, 459
 - declarative markup, 456
 - EditItemTemplate, 459
 - EmptyDataTemplate, 459
 - exercises, 478
 - GroupTemplate, 459, 476
 - InsertItemTemplate, 459
 - ItemSeparatorTemplate, 459
 - ItemTemplate, 455-459
 - LayoutTemplate, 455-459
 - overview, 454-455
 - paging support, 465-467

sorting interface, 464-465
 when to use, 476
Literal Web control
 example, 172-173
 exercises, 190
 ID property, 173
 overview, 171-172
 Text property, 173-177
LiteralTime.aspx web page, 175-177
Locals window (debugger), 102
Location drop-down list (Visual Web Developer), 31
 locations of websites, 61-62
 logging in, 528-530
 logging out, 531-532
 logical entities, 310-311
 logical operators, 334
 Login control, 528-530
 Login.aspx page, 528, 532
 LoginStatus control, 531-532
Long data type, 113
loops
 Do ... Loop, 141-142
 For ... Next, 139-141
 infinite loops, 142
 overview, 139
loose typing, 112

M

magnitude of numbers, 114
MailMessage class, 168
main region (master pages), 550
managing user accounts, 514-515
maps, site maps, 320
masked text boxes, 230-231

master pages, 66
 content pages
 creating, 550-552
 passing information
 between content and
 master pages, 560
 creating, 544
 default content in, 554-555
 default declarative
 markup, 545-546
 definition, 540
 exercises, 562
 inheriting from, 552-553
 nesting, 560
 overview, 506, 539-543
 passing information between
 content and master
 pages, 560
 server-side source
 code, 556-557
 templates, 546-547
 footer region, 550
 headers, 547-548
 main region, 550
 navigation region,
 549-550
 testing, 558
MasterPageFile attribute, 553
math operators, 117-118
MaxLength property (TextBox control), 227-228
membership
 configuring websites to
 support, 510-514
 definition, 510
Menu control, 484
 customizing, 503-504
 displaying site structure
 in, 501
 static versus dynamic
 portions, 501-502

messages, displaying, 574-576
methods. See also
functions; procedures
 calling, 164-165
 constructors
 creating objects with, 162
 default constructors, 163
 definition, 162
 naming conventions, 162
 parameters, 163
 definition, 46-47, 162
 ExecuteReader, 165
 overloaded methods, 163
Microsoft ASP.NET Developer Center, 74
Microsoft Access, 299
Microsoft Internet Information Server (IIS), 23
Microsoft SQL Server, 298
Microsoft SQL Server Management Studio Express Edition, 594-595
Microsoft Visual C# Developer Center, 62
Modify Style dialog box, 548
modularizing control structures
 definition, 142
 functions
 calling, 155
 compared to
 subroutines, 154
 ComputeCostPerMonth,
 151-152
 definition, 142
 parameters, 148
 syntax, 150
 subroutines
 body, 146
 calling, 146
 compared to
 functions, 154

modularizing control structures

- creating, 146
- definition, 142
- example, 142-145
- parameters, 148-150
- ShowWelcomeMessage, 146-150
- mortgage calculator. See financial calculator**
- moving**
 - Visual Web Developer windows, 71-72
 - website files, 67
 - websites, 74
- MSDN Library, 16, 72**
- multiline text boxes, 220-222, 231**
- MultiLineTextBox.aspx page, 220-222**
- MultipleUpdatePanels.aspx web page, 571-573**
- multiplication operator, 117**
- mutable values, 110**
- mutually exclusive buttons, 246**
- MySQL, 299**

N

- Name subproperty (Label control Font property), 178**
- Names subproperty (Label control Font property), 179**
- naming conventions**
 - constructors, 162
 - variables, 110-112
- narrowing casts, 124**
- navigation region (master pages), 549-550**

- navigation. See site navigation**
- nchar data type, 302**
- nested tags (HTML), 29**
- nesting master pages, 560**
- .NET Framework**
 - definition, 8
 - installation, 15-18
- New Web Site command (File menu), 19, 60**
- New Web Site dialog box, 19-20, 31, 60**
- NodeIndent property (TreeView control), 499**
- nodes**
 - site maps, 493
 - TreeView, 497
- NodeStyle property**
 - SiteMapPath control, 494
 - TreeView control, 498
- NodeWrap property (TreeView control), 500**
- NoExpandImageUrl property (TreeView control), 499**
- None value (BorderStyle property), 183**
- nonintegral numeric types, 114-115**
- NoSet value (BorderStyle property), 183**
- null strings, 392**
- numbers**
 - integers, 113-114
 - magnitude, 114
 - nonintegral numeric types, 114-115
- nvarchar data type, 302**

O

- Object data type, 115-116**
- object-oriented programming**
 - definition, 46
 - event-driven programming
 - event handlers, 47-51
 - out-of-order execution, 48-49
 - raising events, 48
 - overview, 46-47
- objects**
 - creating, 162
 - definition, 46, 159-160
 - instantiating, 162
 - instantiation, 88
 - methods. *See methods*
 - Object data type, 115-116
 - object-oriented programming
 - definition, 46
 - event-driven programming, 47-51
 - overview, 46-47
 - properties, 164
 - role of, 161
 - tasks performed by, 161-162
- one-way data binding, 443-444**
- Open Web Site command (File menu), 63**
- Open Web Site dialog box, 63**
- opening websites, 63**
- Operator property (CompareValidator control), 276-277**
- operators**
 - arithmetic operators, 117-118
 - assignment operators, 110-111, 120-122, 127

QuestionLabelText property

- comparison operators, 118-119, 334
- concatenation operator, 119-120
- logical operators, 334
- precedence, 118
- Option Strict On** setting, 126-128, 152
- Options** dialog box (Visual Web Developer), 68-70
- Oracle**, 298
- ORDER BY** clause (SELECT statement), 335-336
- Orientation** property
 - Login control, 530
 - Menu control, 504
- out-of-order** execution, 48-49
- Outset** value (BorderStyle property), 183
- Overline** subproperty (Label control Font property), 179
- overloaded** methods, 163

P

- @Page** directive, 32
- page** numbers, 361
- Page.IsValid** property (RequiredFieldValidator control), 273-275
- PageIndex** property (GridView control), 365
- pager** rows, 361
- PagerSettings** property
 - DetailsView control, 362
 - GridView control, 365
- PagerSettings** property (FormView control), 471
- PagerStyle** property (GridView control), 356
- PagerTemplate** (FormView control), 472
- PageSize** property (GridView control), 365
- Page_Load** event handler, 153
- paging** interface
 - DetailsView control, 362
 - FormView control, 470-471
 - GridView control, 364-366
 - ListView control, 465-467
- parameterized** queries, 342
- parameters**
 - of constructors, 163
 - parameterized queries, 342
 - passing to functions, 148
 - passing to subroutines, 148-150
- parent** nodes (TreeView), 497
- ParentNodeStyle** property (TreeView control), 498
- partial** postback, 564
- passing**
 - information between content and master pages, 560
 - parameters
 - to functions, 148
 - to subroutines, 148-150
- password** text boxes, 222-225
- PasswordLabelText** property (CreateUserWizard), 523
- PasswordRecovery** control, 535
- PasswordRequiredErrorMessage** property (CreateUserWizard), 524
- passwords**, prompting users to enter, 524-525
- PasswordTextBox.aspx** page, 222-225
- PathDirection** property (SiteMapPath control), 493

- PathSeparator** property (SiteMapPath control), 494
- PathSeparatorStyle** property (SiteMapPath control), 494
- pattern** validation, 264
- PerformCalcButton_Click** event handler, 87-88
- populating**
 - drop down lists from databases, 257
 - list Web controls, 411-412
- postback** forms, 199-200, 205, 212
- PostgreSQL**, 299
- precedence** of operators, 118
- PrettyTextBox.aspx** page, 228-229
- primary** key columns, 302-303
- Private** keyword, 154
- programming** languages, 108-109
- progress** messages, displaying, 574-576
- properly** nested tags (HTML), 29
- properties**. *See also specific properties*
 - definition, 46
 - setting, 164
- Protected** keyword, 155
- pseudocode**, 49

Q

- queries**, parameterized, 342. *See also* SQL statements
- Question** property (CreateUserWizard), 523
- QuestionLabelText** property (CreateUserWizard), 523

RadioButton Web control

R

RadioButton Web control

- adding to web pages, 247-249
- definition, 246
- determining when selected, 250-251
- formatting properties, 252
- grouping, 249-250
- GroupName property, 249-250
- mutually exclusive buttons, 246
- Selected property, 250-251
- Text property, 249

RadioButton.aspx page, 247-248

RadioButtonList Web control, 247

- customizing appearance of, 424-425
- enumerating list items, 422
- filtering data with, 426
- overview, 421-422
- SelectedItem and SelectedValue properties, 423-424

raising events, 48

range input validation, 263

RangeValidator control, 282-283

read-only fields, marking, 389-390

rearranging list items, 243

Recent Projects command (File menu), 63

records

- definition, 300
- paging
 - DetailsView control, 362
 - FormView control, 470-471

GridView control, 364-366

ListView control, 465-467

redirect forms, 199-200

redundancy in code, 146-147

refreshing

- pages with postback forms, 212
- with UpdatePanel control, 573

regular expressions, 284

RegularExpressionValidator control, 284-286, 291

RememberMeSet property (Login control), 530

remote websites, visiting, 598-599

renaming website files, 67

RenderCurrentNodeAsLink property (SiteMapPath control), 494

rendered source code, viewing, 92-97

replicating databases, 591-592

- with Database Publishing Wizard, 592-594
- with output scripts, 594-596

required field input validation, 263

RequiredFieldValidator control

- adding to web pages, 267
- client-side validation, 272
- ControlToValidate property, 268-269
- displaying error messages, 270
- Page.IsValid property, 273-275
- server-side validation, 272
- specifying Web control to be validated, 268-269
- summary of features, 275
- testing, 270-272

resizing

- text boxes, 225-226
- Visual Web Developer windows, 71

Ridge value (BorderStyle property), 183

roles (users), 516-517

root nodes

- site maps, 493
- TreeView, 497

RootNodeStyle property

- SiteMapPath control, 494
- TreeView control, 498

RowDeleted event, 384

RowDeleting event, 384

rows. See records

RowStyle property (GridView control), 356

rules, access rules, 517-519

S

sample Web applications, building, 585-589

Sams Teach Yourself HTML and CSS in 24 Hours, 27

Script Debugging Disabled warning, 100

script files, 64

ScriptManager control, 567

scripts, client-side scripts, 272

security for user accounts

- access rules, 517-519
- user roles, 516-517

Select a Master Page dialog box, 550

SELECT statement, 324-325

- DESC clause, 337
- ORDER BY clause, 335-336

- syntax, 328
 - viewing results of, 328-331
 - WHERE clause, 333-335, 440-442, 450
- SelectCommand property** (SqlDataSource control), 327
- selected nodes** (TreeView), 497
- Selected property**
 - drop-down list items, 242
 - RadioButton control, 250-251
- SelectedIndexChanged event** handler, 412-415
- SelectedItem property** (list Web controls), 423-424
- SelectedNodeStyle property** (TreeView control), 498
- SelectedRowStyle property** (GridView control), 356
- SelectedValue property** (list Web controls), 423-424
- sending email**
 - from ASP.NET pages, 168
 - to new user accounts, 525-527
- "Sending Email in ASP.NET"** (article), 535
- sequential execution**, 48
- sequential flow**, 132
- server-side code**, 46
- server-side validation**, 272
- servers**
 - SMTP servers, 520
 - web. See web servers
- serving web pages**
 - dynamic web pages, 11-12
 - static web pages, 10-11
- Short data type**, 114
- shorthand assignment operators**, 121-122, 127
- Show All Genres option** (DropDownList control), 427
- ShowExpandCollapse property** (TreeView control), 500
- ShowFooter property** (GridView control), 355
- ShowHeader property** (GridView control), 355
- ShowLines property** (TreeView control), 500
- ShowWelcomeMessage subroutine**, 146-150
- Silverlight**, 17
- SimpleListView.aspx web page**, 460-461
- SimpleListView2.aspx web page**, 462
- Single data type**, 114
- site map providers**, 506
- site maps**
 - adding to projects, 486-488
 - creating, 488-490
 - default content, 487
 - definition, 320, 483
 - exercises, 507-508
 - nodes, 493
 - overview, 485-486
 - site map providers, 506
- site navigation**
 - definition, 483
 - Menu control, 484
 - customizing, 503-504
 - displaying site structure in, 501
 - static versus dynamic portions, 501-502
 - SiteMapPath control, 484
 - adding to web pages, 491-492
 - customizing, 492-494
 - TreeView control, 484
 - customizing, 497-500
 - displaying site structure in, 496
- siteMapNode element**, 487-490
- SiteMapPath control**, 484
 - adding to web pages, 491-492
 - customizing, 492-494
- Size subproperty** (Label control Font property), 179
- sizing**
 - text boxes, 225-226
 - Visual Web Developer windows, 71
- smart tags**, 321
- SMTP servers**, 520
- SMTP settings** (websites), 519-521
- SmtpClient class**, 168
- Solid value** (BorderStyle property), 183
- Solution Explorer**, 18, 64-67
- SortExpression property** (GridView control), 368
- sorting data**
 - ListView control, 464-465
 - with GridView control, 366-368
 - with SqlDataSource control, 339-340
- source code (ASP.NET)**
 - client-side script code, 64
 - files, 45
 - financial calculator web page, 98-99
 - code listing, 89-91
 - viewing, 92-97
 - writing, 87-88
 - object-oriented programming, 46
 - programming languages, 62-63
 - sequential execution, 48
 - server-side code, 46
 - viewing, 46

Source view

Source view (Visual Web Developer), 30

- description of, 32
- viewing HTML in, 40-45

Split view (Visual Web Developer), 30

SQL (Standard Query Language) statements

- definition, 298, 319
- DELETE, 379
- exercises, 345
- INSERT, 378-379
- overview, 327-328
- retrieving data from multiple tables, 343-344
- SELECT statement, 324-325
 - DESC clause, 337
 - ORDER BY clause, 335-336
 - syntax, 328
 - viewing results of, 328-331
 - WHERE clause, 333-335, 440-442, 450
- UPDATE, 379-380

SQL Server, 8, 16, 298

SQL Server Management Studio Express Edition, 594-595

"SQL Server Views" (article), 306

SqlCommand class, 163-165

SqlDataSource control. *See also* SQL statements

- adding to web pages, 321-322
- configuring data sources, 322-325
- ConnectionString property, 327
- filtering data, 337-339
- ID property, 326
- inserts, updates, and deletions, 374-377

- markup, 342
- SelectCommand property, 327
- sorting data, 339-340
- testing queries, 340-341

SQLite, 299

Start Debugging command (Debug menu), 99

Start Page (Visual Web Developer), 18

start tags (HTML), 26

Start Without Debugging command (Debug menu), 21

starting debugger, 99-102

statements. *See also* loops

- assignment statements, 110
- DELETE, 379
- Dim, 111, 116-117
- If
 - Else clause, 136-137
 - Elseif clause, 137-138
 - example, 133-135
 - syntax, 133

INSERT, 378-379

SELECT, 324-325

- DESC clause, 337
- ORDER BY clause, 335-336
- syntax, 328
- viewing results of, 328-331

- WHERE clause, 333-335, 440-442, 450

UPDATE, 379-380

static web pages

- compared to dynamic web pages, 23
- definition, 9, 64
- serving, 10-11

StaticEnableDefaultPopOutImage property (Menu control), 503

StaticItemFormatString property (Menu control), 504

StaticPopOutImageTextFormatString property (Menu control), 504

StaticPopOutImageUrl property (Menu control), 504

StaticSubMenuIndent property (Menu control), 504

Step keyword, 140

Step Over command (Debug menu), 102

Stop Debugging command (Debug menu), 102

stopping debugger, 102

Strikeout subproperty (Label control Font property), 179

String data type, 115

strings, 115, 302

- blank strings, 392
- concatenation, 119-120
- connection strings
 - definition, 322
 - updating in web.config files, 596-598
- definition, 110
- empty strings, 143
- inserting into other strings, 120
- null strings, 392

strong typing, 113

Structured Query Language. *See* SQL statements

Style Builder dialog box, 548

style sheets, 35, 64

submitting forms, 197

subroutines

- body, 146
- calling, 146
- compared to functions, 154
- creating, 146
- definition, 142
- example, 142-145

parameters, 148-150
 ShowWelcomeMessage,
 146-150
subtraction operator, 117
Suggest (Google), 566

T

<table> element (HTML)
 adding to pages, 33-40
 definition, 33
**Table menu commands, Insert
 Table, 547**

tables
 adding data to, 312-315
 columns
 adding/removing, 316
 Auto-increment
 columns, 303
 bit columns, 431-434
 column types, 301-302
 definition, 300
 primary key columns,
 302-303
 creating, 33-40, 306-308
 definition, 300
 deleting data
 with DELETE
 statement, 379
 with GridView control,
 381-384
 with SqlDataSource
 control, 374-377
 inserting data
 with DetailsView control,
 398-400
 with INSERT statement,
 378-379
 with SqlDataSource
 control, 374-377

records, 300
 updating data
 with DELETE statement,
 379-380
 with GridView control,
 387-397
 with SqlDataSource
 control, 374-377

tags. See elements

templated data Web controls

exercises, 478
 FormView
 available templates, 472
 custom templates,
 473-474
 overview, 469-470
 paging records, 470-471

ListView
 adding templates
 manually, 460-462
 adding templates with
 Configure ListView
 option, 458-460
 AlternatingItemTemplate,
 459
 declarative markup, 456
 EditItemTemplate, 459
 EmptyDataTemplate, 459
 GroupTemplate, 459, 476
 InsertItemTemplate, 459
 ItemSeparatorTemplate,
 459
 ItemTemplate, 455-459
 LayoutTemplate, 455-459
 overview, 454-455
 paging support, 465-467
 sorting interface, 464-465
 when to use, 476
 overview, 453-454

TemplateField, 431, 451

templates

definition, 369, 393
 FormView control
 available templates, 472
 custom templates,
 473-474
 ListView control
 adding templates
 manually, 460-462
 adding templates with
 Configure ListView
 option, 458-460
 AlternatingItemTemplate,
 459
 EditItemTemplate, 459
 EmptyDataTemplate, 459
 GroupTemplate, 459, 476
 InsertItemTemplate, 459
 ItemSeparatorTemplate,
 459
 ItemTemplate, 455-459
 LayoutTemplate, 455-459
 paging support, 465-467
 sorting interface, 464-465
 when to use, 476
 master pages, 546-547
 footer region, 550
 headers, 547-548
 main region, 550
 navigation region,
 549-550
 website templates, 60-61

testing

BMICalculator.aspx page,
 204-206
 CompareValidator control,
 278-279
 with Debug Menu (Visual Web
 Developer), 39-40
 financial calculator web
 page, 91-92

testing

- testing erroneous
 - input, 97-98
 - viewing rendered source code, 92-97
- master pages, 558
- queries with `SqlDataSource` control, 340-341
- `RequiredFieldValidator` control, 270-272
- `SqlDataSource` control, 325
- web pages, 21-22

text

- displaying with `Label` Web control
 - `BackColor` property, 180-182
 - `BorderColor` property, 183
 - borders, 182-184
 - `BorderStyle` property, 182-183
 - `BorderWidth` property, 184
 - colors, 180-182
 - exercises, 190
 - `Font` property, 178-179, 184-185
 - fonts, 178-179, 184-185
 - `ForeColor` property, 180-182
 - `Height` property, 185
 - setting properties programmatically, 188
 - `Text` property, 177
 - `ToolTip` property, 185
 - tooltips, 185
 - `Visible` property, 186
 - `Width` property, 185
- displaying with `Literal` Web control
 - example, 172-173
 - exercises, 190
 - `ID` property, 173

- overview, 171-172
- `Text` property, 173-177

text boxes

- adding to web pages, 216-217
- colors, 228-229
- event handling, 217-219, 231
- exercises, 233-234
- fonts, 228-229
- length, 225-226
- limiting number of characters in, 227-228
- masked text boxes, 230-231
- multiline text boxes, 220-222, 231
- overview, 215
- password text boxes, 222-225

Text property

- drop-down list items, 242
- `Label` control, 177
- `Literal` control, 173-177
- `RadioButton` control, 249

TextBox Web control

- adding to web pages, 81-83, 216-217
- `BackColor` property, 228
- `BorderColor` property, 228
- `BorderStyle` property, 228
- `BorderWidth` property, 228
- colors, 228-229
- `Columns` property, 225-226
- event handling, 217-219, 231
- exercises, 233-234
- `Font` property, 228
- fonts, 228-229
- `ForeColor` property, 228
- limiting number of characters in, 227-228
- masked text boxes, 230-231

- `MaxLength` property, 227-228
- multiline text boxes, 220-222, 231
- overview, 215
- password text boxes, 222-225
- reading values in, 88-89
- resizing, 225-226
- `TextMode` property
 - `MultiLine` value, 220
 - `Password` value, 222

TextBoxPractice.aspx
page, 216-219

TextLayout property (`Login` control), 530

TextMode property (`TextBox` control), 220-222

Toolbox (`Visual Web Developer`), 18, 36-37

ToolTip property (`Label` control), 185

tooltips, 185

TreeView control, 484

- customizing, 497-500
- displaying site structure in, 496

two-way data binding, 443-444

Type property (`CompareValidator` control), 277

types. *See* data types

U

Uls. *See* user interfaces

unary operators, 118

Underline subproperty (`Label` control `Font` property), 179

UPDATE statement, 379-380

user input, validating**UpdatePanel control**

- adding to web pages, 568-570
- definition, 567
- multiple controls, 571-573
- overview, 567
- progress messages, displaying, 574-576
- refreshing, 573

UpdateProgress control, 567**UpdateProgress.aspx web page, 575-576****updating data**

- with GridView control, 387-397
 - custom editing interface, 392-397
 - editing and formatting fields, 391-392
 - null strings/blank strings, 392
 - read-only fields, marking, 389-390
- with SqlDataSource control, 374-377
- with UPDATE statement, 379-380
- web.config files, 596-598

uploading website files, 590-591**user accounts**

- access rules, 517-519
- allowing users to create, 521-525
- anonymous users, 518
- creating, 514-515
- credentials, 510
- customizing, 535
- displaying content based on authentication status, 532-533
- exercises, 537

- inactive user accounts, 527-528
- logging in, 528-530
- logging out, 531-532
- membership
 - configuring websites to support, 510-514
 - definition, 510
- overview, 509-510
- sending email messages to new accounts, 525-527
- user roles, 516-517

user input, collecting

- check boxes
 - adding to web pages, 252-253
 - default selections, 258
 - definition, 252
 - determining which check boxes are selected, 254-255
 - exercises, 259
- drop-down lists
 - adding list items to, 238-239
 - adding to web pages, 239-245
 - copying, 257
 - overview, 238
 - populating from databases, 257
 - properties, 246
 - rearranging order of items, 243
- form element, 196-199
 - example, 198
 - form submission, 197
 - postback forms, 199-200
 - redirect forms, 199-200
- input element, 194-196
- overview, 194

radio buttons

- adding to web pages, 247-249
- definition, 246
- determining when selected, 250-251
- formatting properties, 252
- grouping, 249-250
- mutually exclusive buttons, 246
- text, 249

text boxes

- adding to web pages, 216-217
- colors, 228-229
- event handling, 217-219, 231
- exercises, 233-234
- fonts, 228-229
- length, 225-226
- limiting number of characters in, 227-228
- masked text boxes, 230-231
- multiline text boxes, 220-222, 231
- overview, 215
- password text boxes, 222-225

types of user input, 236-238**Web Forms**

- BMICalculator.aspx example, 202-210
- definition, 201

user input, validating

- CompareValidator control
 - adding to web pages, 275
- comparing input with, 280-282
- ControlToValidate property, 277

user input, validating

ErrorMessage
 property, 277
 Operator property,
 276-277
 testing, 278-279
 Type property, 277
 ValueToCompare
 property, 277
 comparison validation, 263
 CustomValidator control, 289
 data type validation, 263
 definition, 262
 exercises, 293
 formatting properties for
 validation controls,
 286-289
 importance of, 262
 overview, 261
 pattern validation, 264
 range input validation, 263
 RangeValidator control,
 282-283
 RegularExpressionValidator
 control, 284-286, 291
 required field input
 validation, 263
 RequiredFieldValidator control
 adding to web pages, 267
 client-side validation, 272
 ControlToValidate
 property, 268-269
 displaying error
 messages, 270
 Page.IsValid property,
 273-275
 server-side validation, 272
 specifying Web control to
 be validated, 268-269
 summary of features, 275
 testing, 270-272
 sample web page, 265-266

table of validation
 controls, 264-265
 ValidationSummary
 control, 289
user interfaces
 definition, 79
 design requirements, 79-80
 financial calculator web page
 Compute Monthly Cost
 button, 83-84
 Label Web control, 84-86
 overview, 80-81
 PerformCalcButton_Click
 event handler, 87-88
 source code, 87-99
 testing, 91-98
 TextBox Web controls,
 81-83, 88-89
UserName property
 (CreateUserWizard), 523
UserNameLabelText property
 (CreateUserWizard), 523
UserNameRequiredErrorMessage
 property
 (CreateUserWizard), 524
Using the CustomValidator
 Control (article), 289

V

validating input
 CompareValidator control
 adding to web pages, 275
 comparing input
 with, 280-282
 ControlToValidate
 property, 277
 ErrorMessage
 property, 277
 Operator property,
 276-277
 testing, 278-279
 Type property, 277
 ValueToCompare
 property, 277
 comparison validation, 263
 CustomValidator control, 289
 data type validation, 263
 definition, 262
 exercises, 293
 formatting properties for
 validation controls, 286-289
 importance of, 262
 overview, 261
 pattern validation, 264
 range input validation, 263
 RangeValidator control,
 282-283
 RegularExpressionValidator
 control, 284-286, 291
 required field input
 validation, 263
 RequiredFieldValidator control
 adding to web pages, 267
 client-side validation, 272
 ControlToValidate property,
 268-269
 displaying error mes-
 sages, 270
 Page.IsValid property,
 273-275
 server-side validation, 272
 specifying Web control to
 be validated, 268-269
 summary of features, 275
 testing, 270-272
 sample web page, 265-266
 table of validation controls,
 264-265
 ValidationSummary
 control, 289

validation controls

- CompareValidator control
 - adding to web pages, 275
 - comparing input with, 280-282
- ControlToValidate property, 277
- ErrorMessage property, 277
- Operator property, 276-277
- testing, 278-279
- Type property, 277
- ValueToCompare property, 277
- CustomValidator control, 289
- exercises, 293
- formatting properties, 286-289
- RangeValidator control, 282-283
- RegularExpressionValidator control, 284-286, 291
- RequiredFieldValidator control
 - adding to web pages, 267
 - client-side validation, 272
 - ControlToValidate property, 268-269
 - displaying error messages, 270
 - Page.IsValid property, 273-275
 - server-side validation, 272
 - specifying Web control to be validated, 268-269
 - summary of features, 275
 - testing, 270-272
- sample web page, 265-266
- table of, 264-265
- ValidationSummary control, 289

ValidationControlTestBed.aspx page, 265-266

ValidationSummary control, 289

Value property (drop-down list items), 242

values, assigning to variables, 110-111, 116

ValueToCompare property, 277

varchar data type, 302

variables

- assigning values to, 110-111, 116
- data types, 112-113
 - Boolean, 115
 - casting, 123-126
 - DateTime, 115
 - definition, 110
 - Double, 114
 - Integer, 113
 - Long, 113
 - loose typing, 112
 - Object, 115-116
 - Short, 114
 - Single, 114
 - String, 115
 - strong typing, 113
- declaring, 111, 116-117
- definition, 109-110
- naming, 110-112

viewing

- HTML in Visual Web Developer Source view, 40-45
- rendered source code, 92-97
- source code, 46
- Visual Web Developer windows, 71-72

views

- DetailsView. See DetailsView control
- FormView control
 - available templates, 472
 - custom templates, 473-474
 - exercises, 479
 - overview, 469-470
 - paging records, 470-471
- GridView. See GridView control
- ListView control
 - adding templates manually, 460-462
 - adding templates with Configure ListView option, 458-460
 - AlternatingItemTemplate, 459
 - declarative markup, 456
 - EditItemTemplate, 459
 - EmptyDataTemplate, 459
 - exercises, 478
 - GroupTemplate, 459, 476
 - InsertItemTemplate, 459
 - ItemSeparatorTemplate, 459
 - ItemTemplate, 455-459
 - LayoutTemplate, 455-459
 - overview, 454-455
 - paging support, 465-467
 - sorting interface, 464-465
 - when to use, 476
- Visual Web Developer, 30
- Visible property (Label control), 186**
- VisibleWhenLoggedIn property (Login control), 530**
- visiting remote websites, 598-599

Visual Basic

Visual Basic

- classes
 - constructors, 162-163
 - creating, 168
 - definition, 160
 - MailMessage, 168
 - SmtplibClient, 168
 - SqlCommand, 163-165
- control structures
 - definition, 131
 - functions, 142, 148-155
 - If statement, 133-138
 - loops, 139-142
 - overview, 132
 - subroutines, 142-150, 154
- event handlers, 153
- methods
 - calling, 164-165
 - definition, 162
 - ExecuteReader, 165
 - overloaded methods, 163
- objects
 - creating, 162
 - definition, 159-160
 - instantiating, 162
 - properties, 164
 - role of, 161
 - tasks performed by, 161-162
- operators, 117
 - arithmetic operators, 117-118
 - assignment operators, 111, 120-122, 127
 - comparison operators, 118-119, 334
 - concatenation operator, 119-120
 - logical operators, 334
 - precedence, 118

- statements. *See* statements
- strings, 110
- variables
 - assigning values to, 110-111, 116
 - data types, 110-116
 - declaring, 111, 116-117
 - definition, 109-110
 - naming, 110-112

Visual C# Developer Center, 62

Visual Web Developer

- adding content to websites, 64-67
- AutoRecover feature, 69
- creating web pages, 30-33
- creating websites, 19-21
 - locations, 61-62
 - source code programming languages, 62-63
 - templates, 60-61
- customization, 68-71
- Debug menu, 39-40
- definition, 8
- deleting files, 67
- Design view, 30, 33-40
- exercises, 56-57, 76
- help, 72-73
- Image Web controls, 37
- installation, 15-18
- Location drop-down list, 31
- moving files, 67
- opening websites, 63
- overview, 59
- renaming files, 67
- Solution Explorer, 18, 64-67
- Source view, 30-32, 40-45
- Split view, 30
- Start Page, 18
- Toolbox, 18, 36-37

- windows
 - closing, 71
 - moving, 71-72
 - resizing, 71
 - viewing, 71-72

W-X-Y-Z

“Walkthrough: Debugging Web Pages in Visual Web Developer” (article), 102

Watch window (debugger), 102

web browsers, AJAX compatibility, 577

Web controls. *See also specific Web controls*

- adding to web pages, 37-40
- choosing, 237-238
- data bindings, specifying, 444-449
- definition, 30
- list Web controls
 - automatic postback, 415
 - binding data to, 409-410
 - CheckBoxList, 421-425
 - definition, 408
 - DropDownList, 416-420, 427
 - exercises, 428
 - overview, 408-409
 - populating, 411-412
 - programmatically responding to changed selections, 412-415
 - RadioButtonList, 421-425
 - programmatically working with, 51-54

Web Forms

BMICalculator.aspx
 example, 202-203
 processing, 206-208
 testing, 204-206
 writing code for, 208-210

definition, 81, 201

web pages. See also websites

AccessingData.aspx,
 321, 337
 AJAXSimple.aspx, 568-570
 CheckBox.aspx, 252-253
 content pages
 creating, 550-552
 passing information
 between content and
 master pages, 560
 CreateAccount.aspx, 521-522
 creating, 21-22, 30-33
 deployment, 14
 design requirements
 features, 78-79
 overview, 77-78
 user interfaces, 79-80
 DetailsView.aspx, 360
 displaying data with data
 Web controls
 DetailsView control,
 360-363, 369-371
 GridView control,
 351-360, 364-371
 overview, 347-351
 DropDownList.aspx,
 239, 244-245
 dynamic. *See* dynamic
 web pages
 financial calculator web page
 Compute Monthly Cost
 button, 83-84
 design requirements,
 78-80

Label Web control, 84-86
 overview, 80-81

PerformCalcButton_Click
 event handler, 87
 source code, 87-99
 testing, 91-98

TextBox Web controls,
 81-83, 88-89

hosting

through ASP.NET
 Development Web
 Server, 13-14

through web-hosting
 companies, 14

ListControls.aspx, 409-410

LiteralTime.aspx, 175-177

Login.aspx, 528, 532

master pages, 66

 creating, 544-546

 default content
 in, 554-555

 definition, 540

 exercises, 562

 inheriting from, 552-553

 nesting, 560

 overview, 506, 539-543

 passing information
 between content and
 master pages, 560

 server-side source
 code, 556-557

 templates, 546-550

 testing, 558

MultiLineTextBox.aspx,
 220-222

MultipleUpdatePanels.aspx,
 571-573

PasswordTextBox.aspx,
 222-225

PrettyTextBox.aspx, 228-229

RadioButton.aspx, 247-248

sending email from, 168

SimpleListView.aspx,
 460-461

SimpleListView2.aspx, 462

source code

 executing, 11

 files, 45

 server-side code, 46

 viewing, 46

static web pages

 compared to dynamic web
 pages, 23

 definition, 9, 64

 serving, 10-11

testing, 21-22

TextBoxPractice.aspx,
 216-219

UpdateProgress.aspx,
 575-576

user input, collecting

 check boxes, 252-255,
 258-259

 drop-down lists,
 238-246, 257

 form element, 196-200

 input element, 194-196

 overview, 194

 radio buttons, 246-252

 text boxes. *See* text boxes

 types of user input,
 236-238

 Web Forms. *See*
 Web Forms

user input, validating

 CompareValidator
 control, 275-282

 comparison
 validation, 263

 CustomValidator
 control, 289

 data type validation, 263

web pages

- definition, 262
- exercises, 293
- formatting properties
 - for validation controls, 286-289
- importance of, 262
- overview, 261
- pattern validation, 264
- range input
 - validation, 263
- RangeValidator control, 282-283
- RegularExpressionValidator control, 284-286, 291
- required field input
 - validation, 263
- RequiredFieldValidator control, 267-275
- sample web
 - page, 265-266
- table of validation controls, 264-265
- ValidationSummary control, 289
- ValidationControlTestBed.aspx, 265-266
- web requests, 10**
- web servers**
 - ASP.NET Development Web Server, 13-14
 - definition, 10
 - hosting web pages
 - through ASP.NET Development Web Server, 13-14
 - through web-hosting companies, 14
 - IIS (Internet Information Server), 23
 - serving web pages
 - dynamic web pages, 11-12
 - static web pages, 10-11
- Web Site Administration Tool, 600**
- web-hosting companies, 14, 74, 582-583**
- Web.config file, 31, 64, 596-598**
- Web.sitemap file, 486**
- websites. See also web pages**
 - Amazon.com, 8
 - creating, 19-21
 - locations, 61-62
 - source code programming languages, 62-63
 - templates, 60-61
 - deployment
 - building sample Web applications, 585-589
 - deployment process, 584-585
 - hosting websites
 - from personal computers, 584
 - overview, 581
 - replicating databases, 591-596
 - updating web.config settings, 596-598
 - uploading website files, 590-591
 - visiting remote website, 598-599
 - web-hosting companies, 582-583
- files
 - adding, 64-67
 - deleting, 67
 - file types, 63-64
 - moving, 67
 - renaming, 67
- hosting, 61
 - from personal computers, 584
 - web-hosting companies, 74, 582-583
- moving, 74
- opening with Visual Web Developer, 63
- site navigation
 - definition, 483
 - Menu control, 484, 501-504
 - SiteMapPath control, 484, 491-494
 - TreeView control, 484, 496-500
- sitemaps. *See* sitemaps
- STMP settings, 519-521
- user accounts. *See* user accounts
- WHERE clause (SELECT statement), 333-335, 440-442, 450**
- whitespace in HTML, 27-29**
- widening casts, 124**
- Width property (Label control), 185**
- wildcards, 440-442, 450**
- windows, Visual Web Developer windows**
 - closing, 71
 - moving, 71-72
 - resizing, 71
 - viewing, 71-72
- Windows authentication, 511**
- wizards**
 - Configure Data Source Wizard, 322-325
 - Database Publishing Wizard, 592-594
- "Writing a Stored Procedure" (article), 306**
- XML files**
 - definition, 320, 484
 - siteMapNode element, 487-490