

Scott Mitchell



STARTER KIT

DVD includes
Visual Web Developer
2010 Express and
ASP.NET 4

Sams **Teach Yourself**

ASP.NET 4

in **24**
Hours

SAMS

Sams Teach Yourself ASP.NET 4 in 24 Hours, Complete Starter Kit

Copyright © 2010 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: 978-0-672-33305-7

ISBN-10: 0-672-33305-8

Library of Congress Cataloging-in-Publication Data:

Mitchell, Scott, 1978-

Sams teach yourself ASP.NET 4 in 24 hours complete starter kit / Scott Mitchell. — 1st ed.
p. cm.

Includes bibliographical references and index.

ISBN 978-0-672-33305-7 (alk. paper)

1. Active server pages. 2. Microsoft .NET. 3. Web sites—Design. I. Title. II. Title: Teach yourself ASP.NET 4 in 24 hours complete starter kit. III. Title: ASP.NET 4 in 24 hours complete starter kit.

TK5105 .8885.A26M585 2010

006.7'882—dc22

2010016855

Printed in the United States of America

First Printing June 2010

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 as 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 of 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

Lori Lyons

Copy Editor

Bart Reed

Indexer

Brad Herriman

Proofreader

Kay Hoskin

Technical Editor

Eric Weinberger

Publishing Coordinator

Cindy Teeters

Multimedia Developer

Dan Scherf

Book Designer

Gary Adair

Compositor

Nonie Ratcliff

Introduction

My first exposure to the World Wide Web came at college during the latter half of the 1990s. At the time, you had to be a hard-core computer programmer to build even the most rudimentary website. Today, with technologies like Microsoft's ASP.NET, creating powerful, data-driven, dynamic web applications couldn't be easier. Over the course of 24 hours you will see just how fun and easy it can be to build useful, real-world websites using ASP.NET.

ASP.NET web applications are composed of individual ASP.NET web pages. As we will see in numerous examples, 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.

Visual Web Developer is a free development editor used to create and test ASP.NET pages, and 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.

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 4,” 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, “Introducing 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 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 21, “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?

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

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*. 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 4

In this hour, we will cover

- ▶ What ASP.NET is
- ▶ The system requirements for using ASP.NET
- ▶ The software that must be installed prior to using ASP.NET
- ▶ Installing the .NET Framework, Visual Web Developer, and SQL Server 2008
- ▶ Taking a quick tour of Visual Web Developer
- ▶ Creating a simple ASP.NET 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 sign in, the front page of Amazon.com shows recommended products based on your previous purchases. This is a dynamic web page because it is a single web page whose content is customized based on who is visiting. This book explores how to create such dynamic web pages using ASP.NET.

ASP.NET is a robust and mature technology. 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. And ASP.NET 4 was unveiled in April 2010.

Before we create our first ASP.NET website, we need to install the .NET Framework, Visual Web Developer, and SQL Server 2008. 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, meaning that you can create them using any text editor (such as Microsoft Notepad), but if you've created websites before, you know that using a tool such as Microsoft Expression Web or Adobe Dreamweaver makes the development process much easier than using a generic text editor such as Notepad. This is the case for ASP.NET, as well.

SQL Server 2008 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, "Introducing Databases," we'll see how to create, query, and modify databases from an ASP.NET page.

This hour focuses on installing the necessary software so that we can start creating ASP.NET web applications. 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 such as 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 the 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 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 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.

It is important to understand the differences between how a website serves static web pages versus dynamic web pages.

ASP.NET is only one of many technologies that can be employed to create dynamic web pages. Other technologies include ASP—ASP.NET’s predecessor—PHP, JSP, and ColdFusion. If you have experience developing web applications with other web programming technologies, you may already be well versed in the material presented in the next two sections. If this is the case, feel free to skip ahead to the “Installing the .NET Framework, Visual Web Developer, and SQL Server 2008” section.

**By the
Way**

**Did you
Know?**

Serving Static Web Pages

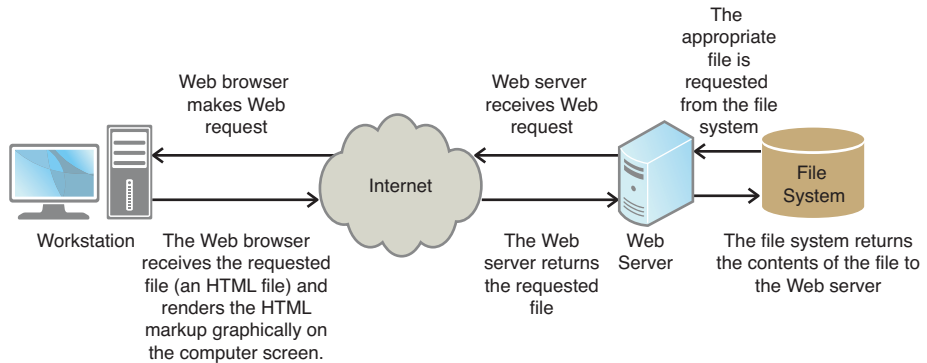
If you’ve developed websites before, you likely know that a website requires a **web server**.

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 request.

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).

FIGURE 1.1

The web server handles incoming web requests.

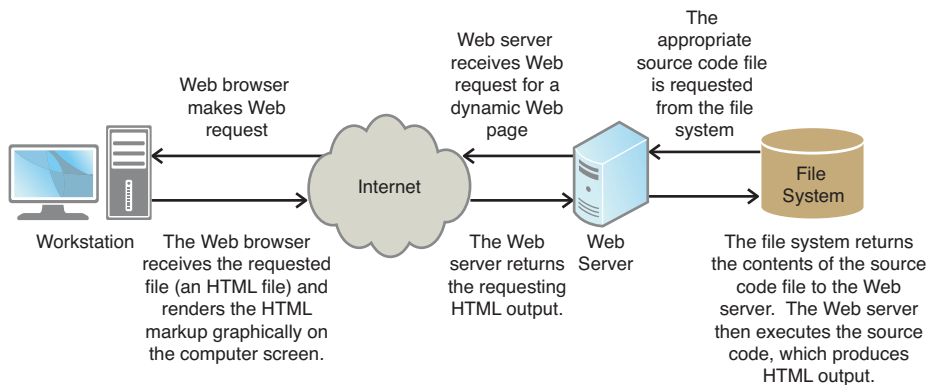


Amazon.com's web server determines what file corresponds to the requested URL and returns the contents of this file to your browser.

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

To accommodate dynamic content, dynamic web pages contain source code that is **executed** when the page is requested (see Figure 1.2). The executing code produces the HTML that is 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.

With this model, 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 dynamic page on August 1, 2010, 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: August 1, 2010, 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 August 1, 2008, 4:15:10 p.m.

Figure 1.2 is, in actuality, a slightly oversimplified model. Commonly, the web server and the software that executes the dynamic web page's 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 page—the web server hands off the responsibility of executing the page to the **ASP.NET engine** (see Figure 1.3).

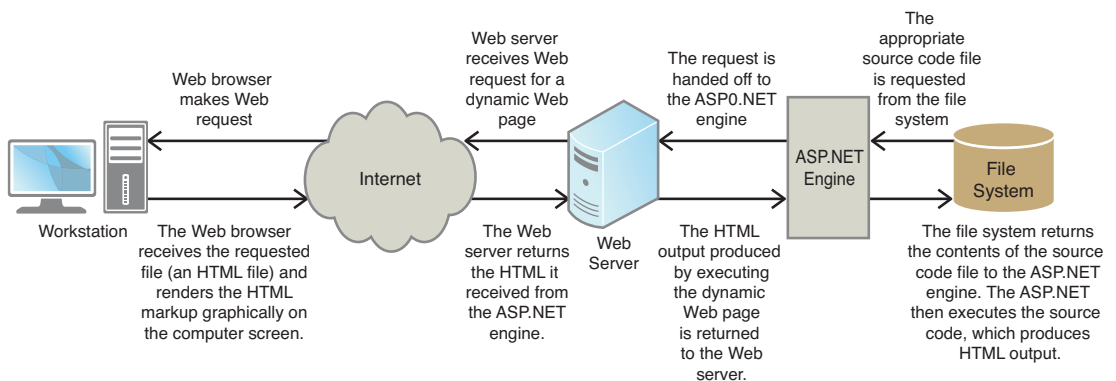


FIGURE 1.3
Execution of an ASP.NET page is handled by the ASP.NET engine.

A common way that web servers determine whether the requested page is a dynamic or static web page is by the requested file's extension. For example, if the extension is `.aspx`, the web server knows the request is for an ASP.NET 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 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 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 Pages

To view an ASP.NET web page that resides on a web server, we need to request it using a web browser. The browser sends a request to the web server, which then dispatches the request to the ASP.NET engine. The ASP.NET engine processes the requested page, returns the resulting HTML to the web server, which then sends it back to the browser, where it is displayed to the user. When you're developing ASP.NET websites, the ASP.NET pages you create are saved on your personal computer. 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. 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 following 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. Hour 24, "Deploying Your Website," explores how to move an ASP.NET website from your personal computer to a web-hosting company's computers. After a website has been successfully deployed to a web-hosting company, you, or anyone else on the Internet, can visit the site.

The examples and lessons presented in Hours 1 through 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.

**By the
Way**

Installing the .NET Framework, Visual Web Developer, and SQL Server 2008

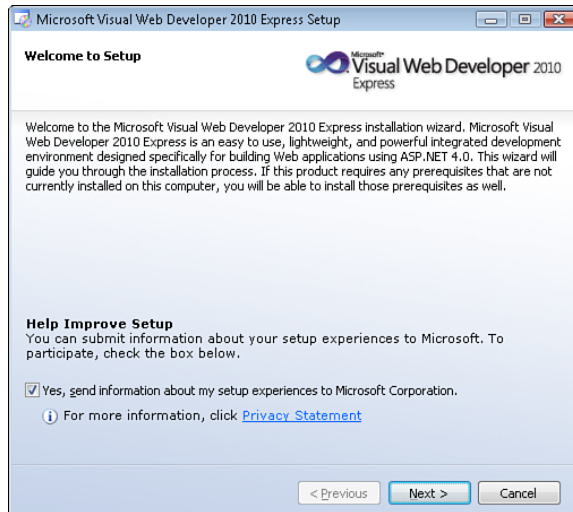
Three components need to be installed before we can start building ASP.NET applications:

- ▶ **The .NET Framework**—Contains the ASP.NET engine, which is used to handle requests for ASP.NET pages. To install the .NET Framework engine, your computer must be running Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, or Windows 7.
- ▶ **Visual Web Developer**—The tool of choice for creating, editing, and testing ASP.NET pages.
- ▶ **SQL Server 2008**—A powerful database engine that is used extensively from Hour 14, “Accessing Data with the Data Source Web Controls,” onward.

The book’s CD includes the installation program for Visual Web Developer. Because Visual Web Developer is designed for developing ASP.NET websites, installing it automatically installs the .NET Framework and other required ASP.NET tools. You can also optionally install SQL Server 2008.

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.

FIGURE 1.4
Start the installation process by inserting the CD into your computer’s CD-ROM drive.



Click the Next button twice to advance past the welcome and license terms screens. At this point you should see the screen shown in Figure 1.5. From here, you can choose whether to install Microsoft SQL Server 2008 Express Edition. Installing SQL Server 2008 is optional in the sense that Visual Web Developer will install successfully

with or without it; however, the latter half of this book's examples rely on SQL Server 2008 being installed. Therefore, make sure that this check box is selected.

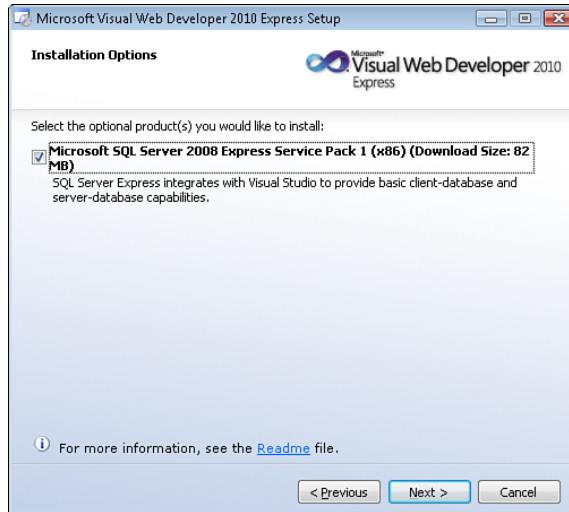


FIGURE 1.5
Make sure to install SQL Server 2008 Express Edition.

The “Installation Options” screen shown in Figure 1.5 is only displayed if you do not have SQL Server 2008 Express Edition already installed on your computer. If you already have this software installed, you will be taken directly to the step shown in Figure 1.6.

**By the
Way**

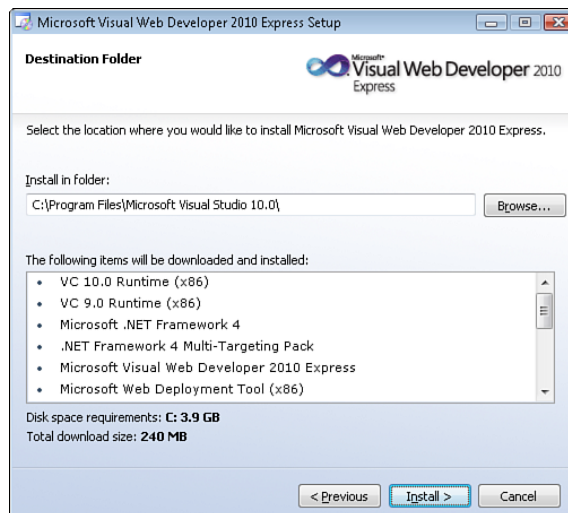
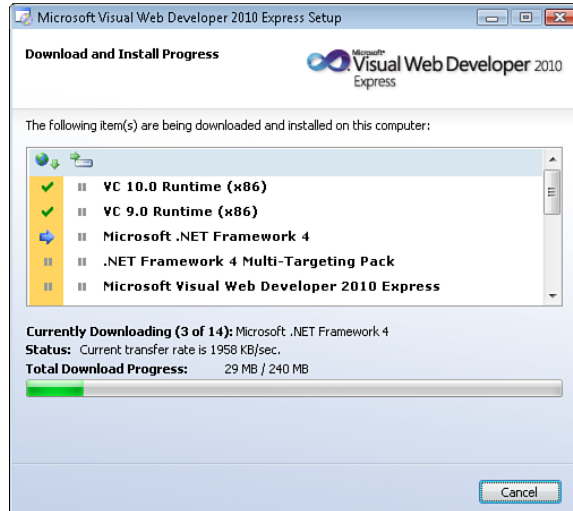


FIGURE 1.6
Specify the folder in which to install Visual Web Developer.

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 1.7 Monitor the installation's progress.



A Brief Tour of Visual Web Developer

When the installation process completes, take a moment to give Visual Web Developer a test spin. To launch Visual Web Developer, go to the Start menu, choose Programs, and click Microsoft Visual Web Developer 2010 Express. Figure 1.8 shows Visual Web Developer after it has loaded.

When you open Visual Web Developer, the Start Page is initially shown. This Start Page includes a list of recent projects on the left and a Get Started section on the right that includes links for accomplishing common tasks. There's also a Latest News tab on the Start Page that lists recent developer articles from Microsoft's website.

On the far left you'll find tabs for the Toolbox, CSS Properties, and Manage Styles windows. You can expand these windows by hovering your mouse over the tabs. At this point, these three windows are empty, but when you're creating or editing an ASP.NET page they'll include a variety of options. For example, the Toolbox is where you'll find the plethora of ASP.NET Web controls that can be added to an ASP.NET page. (We'll discuss what Web controls are and their purpose in the next hour.) The CSS Properties and Manage Styles windows are used to define style and appearance settings for the HTML and Web control elements within a web page.

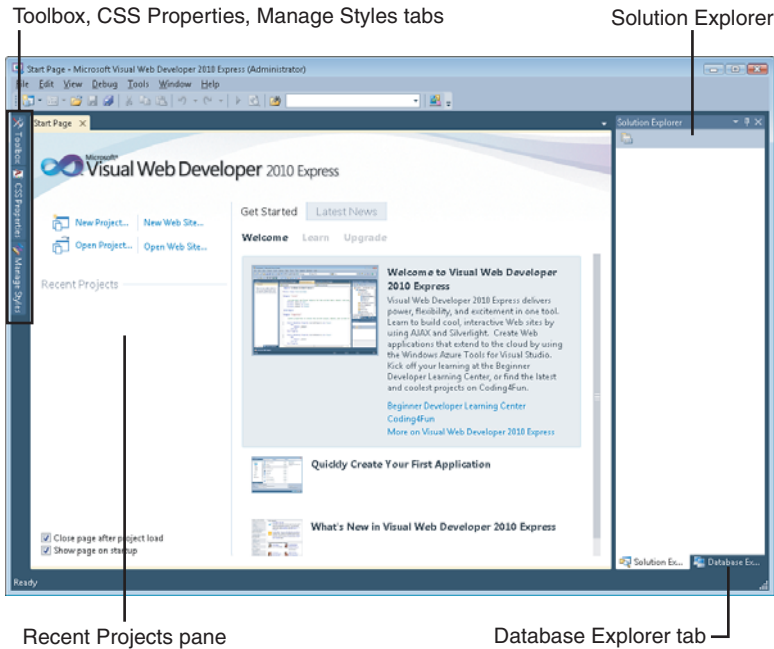


FIGURE 1.8
The Start Page shows when Visual Web Developer is loaded.

On the right of the screen, you'll find the Solution Explorer. From the Start Page the Solution Explorer 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. 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.

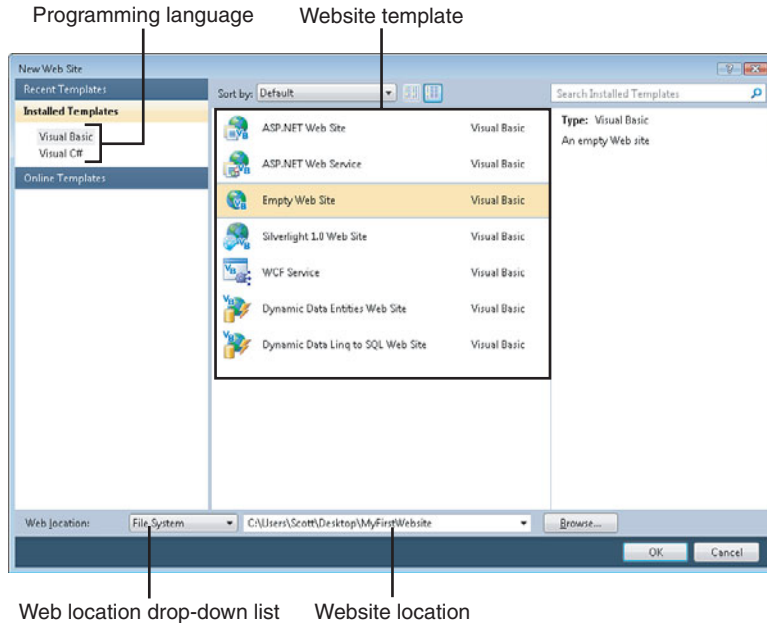
Creating a New ASP.NET Website

To create and design an ASP.NET page, we must first create an ASP.NET website. There are several ways to create a new ASP.NET website from Visual Web Developer. 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 New Web Site link above the Recent Projects pane of the Start Page.

All these approaches bring up the New Web Site dialog box, which is 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." By default, the New Web Site dialog box should have Visual Basic as the selected programming language, ASP.NET Web Site as the selected website template, and the File System option selected in the Web location drop-down list. Change the website template from ASP.NET Web Site to Empty Web Site. Next, change the location of

the website so that it will be created in a folder named MyFirstWebsite on your desktop. You can type in the name or click the Browse button to select the folder.

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.

Watch Out!

When you are creating a new website using the Empty Web Site template, Visual Web Developer adds a single configuration file to the project named `web.config`. Using other templates causes Visual Web Developer to add many more files and folders to the new website. For example, if you selected the ASP.NET Web Site template, your new website would have folders named `Account`, `App_Data`, `Scripts`, and `Styles`, along with files such as `About.aspx`, `Default.aspx`, `Global.asax`, and others. These added files can be helpful if you are already familiar with ASP.NET, but can be distracting when learning about ASP.NET.

If you accidentally selected a template other than the Empty Web Site template, go to the File menu, choose New Web Site, and create a new website using the correct template.

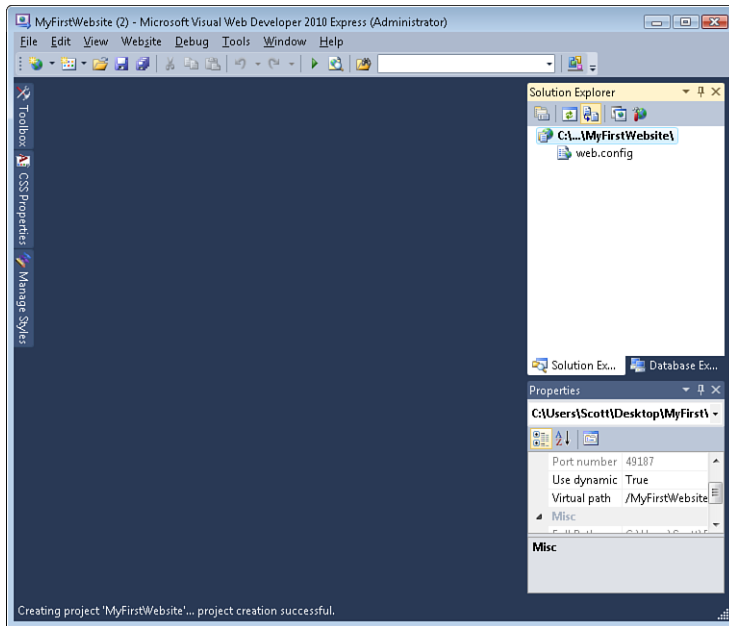


FIGURE 1.10
A new website
has been created
that contains a
single file,
web.config.

Creating a Simple ASP.NET Page

When you fire up your browser and visit a website, your browser requests a particular web page from the web server. At the moment, our website does not yet contain any ASP.NET pages, which means there is no way to visit the site. Let's add an ASP.NET page to our website.

Right-click the website name in the Solution Explorer and choose the Add New Item menu option. This brings up the Add New Item dialog box shown in Figure 1.11. The default settings add a new ASP.NET page to the project named `Default.aspx`. We'll examine this dialog box and its various options in Hour 3. For now, make sure that the Visual Basic programming language and Web Form item templates are selected. Ensure that the name is set to `Default.aspx` and that the Place code in separate file check box is checked, and then click the Add button.

Figure 1.12 shows Visual Web Developer after the `Default.aspx` page has been added to the website. Note that `Default.aspx` is listed in the Solution Explorer and that its contents are displayed in the main window.

FIGURE 1.11
Add a new ASP.NET page to the website.

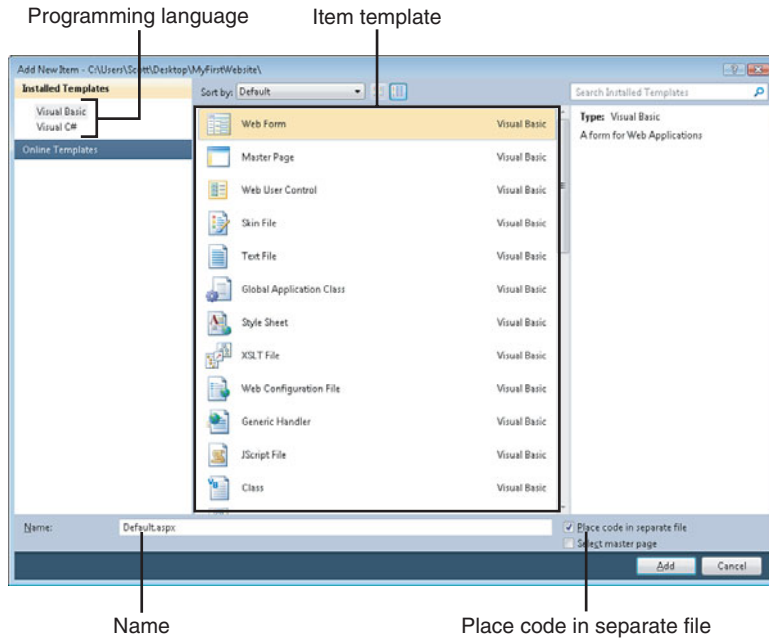
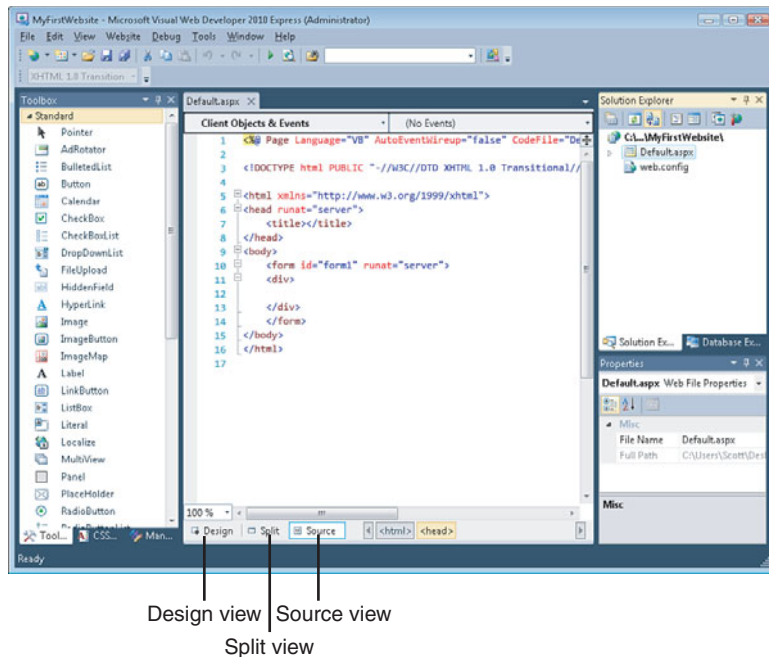


FIGURE 1.12
Visual Web Developer displays the contents of the Default.aspx file.



Right now `Default.aspx` 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 the source code. If you expand `Default.aspx` in the Solution Explorer, you'll see that there's another file, `Default.aspx.vb`. 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**

There are three "views" from which you can work with the HTML elements and Web controls in an ASP.NET page. The first is the Source view, which shows the page's HTML markup and Web control syntax. This is the default view and the one shown in Figure 1.12. 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 or by typing Shift-F7 on your keyboard.

You can reposition the Properties, Toolbox, Solution Explorer, Database Explorer, and other windows by clicking their title bars and dragging them elsewhere, or you can 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?**

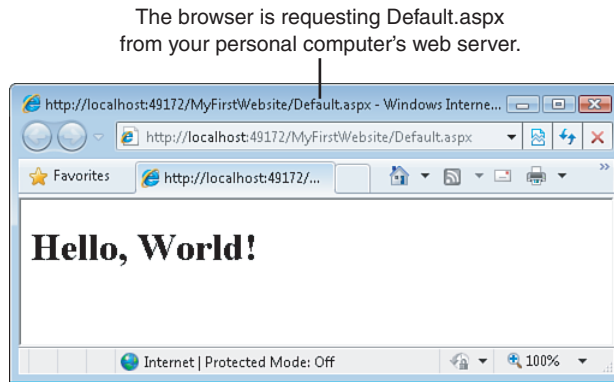
Testing the ASP.NET Page

For us to view or test an ASP.NET 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 Solution Explorer, right-click `Default.aspx`, and choose the View in Browser 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 1.13). The *portNumber* portion in the URL will depend on the port selected by the ASP.NET Development Web Server.

FIGURE 1.13
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.

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 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 .NET Framework, Visual Web Developer, and SQL Server 2008" section, we looked at how to install Visual Web Developer, along with the .NET Framework and SQL Server 2008. In the "A Brief Tour of Visual Web Developer" section, we poked around Visual Web Developer and created our first ASP.NET 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, database 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 pages locally. The computers at 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, Windows 7, 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 pages can include Visual Basic or C# source code.
4. What software packages must be installed to serve ASP.NET pages from a computer?
5. When should you consider using a web-hosting company to host your ASP.NET website?

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 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 page to be dynamically generated for each request.
3. True.
4. For a computer to serve ASP.NET 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 pages.

Index

@Page directive, 31-32
4GuysFromRolla.com, 71

A

access modifiers, keywords, 143
access rules, user accounts, 529-532
AccessDataSource Web control, 307
accounts. *See* user accounts
Add New Item dialog box, 62-63, 290, 472
Add WHERE Clause dialog box, 324-326
adding templates, ListView Web control, 445-450
addition (+) operator, 110
Advanced SQL Generation Options dialog box, 360
Ajax, 555-558, 568
 ASP.NET Ajax Control Toolkit, 569
 benefits, 562-563
 browser compatibility, 568
 Google Gmail, 557
 library, 558
 loading images, 568
 partial page postbacks, 556-557
 progress messages for, 566-567
ScriptManager Web control, 558, 560
Timer control, 569
UpdatePanel Web control, 558-559
 adding to Web pages, 559-562
 multiple controls, 563-565
 user action refresh, 564
UpdateProgress Web control, displaying progress messages, 566-567

AllowPaging property (DetailsView), 349
AllowPaging property (FormView), 458-459
AllowPaging property (GridView), 352

AlternatingItemTemplate (ListView)

- AlternatingItemTemplate (ListView)**, 446
 - AlternatingRowStyle property (GridView)**, 342
 - Amazon.com**, 8-9
 - ampersand (&) concatenation operator**, 112
 - anonymous users**, 530
 - Answer property (CreateUserWizard)**, 536
 - AnswerLabelText property (CreateUserWizard)**, 536
 - apostrophe (')**, Visual Basic components, 92
 - appearance**
 - customizing
 - SiteMapPath, 478-481
 - TreeView, 483-486
 - DetailsView Web control, customizing, 350-351
 - GridView Web control, customizing, 339-347, 356
 - applications, Web applications, building**, 577-581
 - arithmetic operators**, 110-111
 - ASP.NET**, 8-9
 - online resources, 71
 - ASP.NET 3.5 Unleashed and Create Your Own Website**, 322
 - ASP.NET Ajax Control Toolkit**, 569
 - ASP.NET Ajax Library**, 558
 - ScriptManager Web control, 558-560
 - UpdatePanel Web control, 558-559
 - adding to Web pages, 559-562
 - multiple controls, 563-565
 - user action refresh, 564
 - UpdateProgress Web control, 558-559
 - displaying progress messages, 566-567
 - ASP.NET Data Web Controls**, 246
 - ASP.NET engine**, 11-12
 - ASP.NET web pages**. See websites
 - ASP.NET Web Site Administration Tool**, 522-523, 592
 - ASP.NET Web Site template**, 547-549
 - ASPNETDB database**, 524
 - ASPNETDB.MDF database**, 525
 - .aspx file extension**, 12
 - assignment (=) operators**, 104, 113-115
 - shorthand, 114-115, 118
 - assignment statements**, 104-105
 - assignments, variable declaration, performing**, 110
 - AssociatedUpdatePanelID property (UpdateProgress)**, 567
 - asterisk (*), SELECT statement**, 315
 - authentication**
 - displaying content based on, 545-547
 - forms-based authentication, 523
 - Windows authentication, 523
 - Auto Format smart tag (FormView)**, 457
 - auto-increment columns**, 289
 - AutoFormat dialog box**, 480-481
 - GridView Web control, formatting, 346-347
 - AutoGenerateColumns property (GridView)**, 339
 - AutoPostBack property (DropDownList)**, 406
 - AutoPostBack property (list Web controls)**, 401
 - AutoRecover feature (Visual Web Developer)**, 66
- ## B
- BackColor property (DropDownList)**, 232
 - BackColor property (GridView)**, 341
 - BackColor property (Label)**, 166-167
 - BackColor property (SiteMapPath)**, 479
 - BackColor property (TextBox)**, 214-216
 - BackColorUrl property (GridView)**, 341
 - Beighley, Lynn**, 298
 - binary operators**, 110
 - binding data, list Web controls**, 395-397
 - Bit column**, 287
 - bit columns, displaying**, 418-420
 - block elements (HTML)**, 28
 - Body Mass Index (BMI) calculator**, 180, 192-194
 - form element, 182-187
 - input element, 180-182

- testing, 190-192
 - Web Form, 188-190
 - writing, 195-197
 - BodyFileName** property (MailDefinition), 539
 - Bold** subproperty (Label control Font property), 164
 - bold** tag (HTML), 26
 - Book of Basketball, The*, 301
 - Books** table, adding data to, 298-301
 - BooksDataSource** Web control, 442
 - Boolean** data types, 108
 - border** properties, Label Web control, 168-170
 - BorderColor** property (GridView), 341
 - BorderColor property (SiteMapPath), 479
 - BorderColor property (TextBox), 214
 - borders**, Label property, 169
 - BorderStyle** property (GridView), 341
 - (SiteMapPath), 479
 - (TextBox), 214
 - BorderWidth** property (GridView), 341
 - (SiteMapPath), 479
 - (TextBox), 214
 - BoundField**, 417
 - breadcrumbs**, 470
 - displaying, SiteMapPath Web control, 476-478
 - break** tag (HTML), 27
 - breakpoints**, setting, 94-96
 - Browser** Receives Elements listing (9.4), 191
 - browsers**, Ajax, compatibility, 568
 - ButtonField**, 417
 - buttons**, web pages, adding to, 79-80
 - ByRef** keyword, 137
 - ByVal** keyword, 137
- C**
- calculator web page. See financial calculator web page
 - calling
 - functions, 143
 - methods, Visual Basic objects, 149, 152-153
 - cascading style sheets (CSS)**, 36
 - casting** data types, 116-117
 - Catcher in the Rye, The*, 301
 - CausesValidation** property, validation Web control, 276
 - ChangePassword** Web control, 550
 - char** data type, 288
 - characters**, text boxes, limiting, 213-214
 - check** boxes, web pages, 238-242
 - determining checks, 239-242
 - CheckBox** Web control, 238-242
 - determining checks, 239-242
 - CheckBoxField**, 417-420
 - CheckBoxList** Web control, 406-407
 - customizing appearance, 410-411
 - Items property, 407-409
 - SelectedItem property, 408-410
 - SelectedValue property, 408-410
 - Choose a Genre** option, DropDownList Web control, 412
 - Choose Data Source** smart tag (FormView), 457
 - Choose Location** dialog box, 59-60
 - classes**, 45, 148
 - constructors, 150-151
 - creating, 155
 - definition, 148
 - MailMessage, 155
 - objects, 147-149
 - overloaded, 151
 - SmtpClient, 155
 - SqlCommand, 151
 - classifications**
 - user input, 222-223
 - users by roles, 528-529
 - Click** event, Button Web controls, 83
 - Click Event Handler** listing (10.1), 205
 - client-side** script, 61, 258
 - client-side** validation, 258
 - closing** Visual Web Developer windows, 69

closing tag (HTML)**closing tag (HTML), 26****code**

BMICalculator.aspx, writing, 195-197

client-side script code, 61, 258

master pages, 513-516

sequential flow, 122

server-side code, 91

source code

event handlers, 48-49

examining, 44-45

OOP (object-oriented programming), 45-48

web controls, 49-52

web pages

debugging, 93-96

examining, 92-93

writing for, 82-86

Code for the BMI Calculation listing (9.6), 195**Code for the Button's Click Event Handler listing (11.1), 237****code listings**

2.1 (Snippet of HTML with Extraneous Whitespace), 27

2.2 (Markup Generated by the Designer), 40

2.3 (HTML Received by the Browser When Visiting Default.aspx), 42

2.4 (Page_Load Event Handler Fires in Response to the Page's Load Event), 49

4.1 (PerformCalcButton Button's Click Event Handler), 85-86

4.2 (HTML Markup Received by the Browser When First Visiting the Page), 87

4.3 (HTML Markup Received by the Browser After the User Has Clicked), 89

6.1 (Different Message Is Displayed Based on the Current Hour), 124-126

6.2 (Page_Load Event Handler), 132

6.3 (Message Is Displayed Four Times when the Button Web Control Is), 133

6.4 (Code to Display the "Welcome to my Website" message), 135

6.5 (ShowWelcomeMessage Subroutine Accepts a Parameter), 138

6.6 (ComputeCostPerMonth Function), 139

6.7 (PerformCalcButton_Click Event Handler Calls the ComputeMonthlyCost Function), 140

8.1 (CurrentTime Literal Control), 162

9.1 (First Draft of the BMI Calculator), 181

9.2 (Second Draft of the BMI Calculator), 182

9.3 (<form> Element has been Added), 184

9.4 (Browser Receives Elements), 191

9.5 (HTML Received by the Browser), 193

9.6 (Code for the BMI Calculation), 195

10.1 (Click Event Handler), 205

11.1 (Code for the Button's Click Event Handler), 237

11.2 (User's Ice Cream Preferences Are Displayed), 241

15.1 (GridView's Markup), 338

16.1 (SqlDataSource Control's Markup), 362

16.2 (TemplateField Has an ItemTemplate), 381

18.1 (Genre TemplateField's Declarative Markup), 435

19.1 (Simple ListView Control's Declarative Markup), 444

19.2 (ListView Uses Two Templates), 449

20.1 (Default Content of the Site Map), 473

20.2 (Completed Site Map Contents), 476

21.1 (Default Declarative Markup for a Master Page), 501

21.2 (Markup of a Content Page), 508

21.3 (Master Page Has Two Event Handlers), 514

- 22.1 (Plain-Text Email Includes the User's Username), 538
- 22.2 (HTML-Formatted Email), 538
- 23.1 (Page Includes Two Labels), 561
- code redundancy, reducing, subroutines and functions, 135-136
- Code to Display the "Welcome to my Website" listing (6.4), 135
- CodeDemo.master, 513
- coercion. See implicit casting
- CollapseImageTooltip property (TreeView), 485
- CollapseImageUrl property (TreeView), 485
- collecting user input, text boxes, 201-216
- colors
 - HTML, 167
 - in text boxes, 214-215
 - text boxes, changing, 214-216
 - Visual Web Developer, 67
- columns, 285-288
 - adding/removing, 302
 - auto-increment columns, 289
 - Bit, 287
 - datetime, 287
 - float, 287
 - Int, 287
 - money, 287
 - nvarchar(n), 287
 - primary key columns, 288-290
 - variable length character columns, 288
- Columns property (TextBox), 212-214
- CommandArgument property (ListView), 452
- CommandField, 417
- CommandName property (ListView), 452
- commands
 - Debug menu
 - Continue, 96
 - Step Over, 96
 - File menu
 - New Web Site, 58
 - Open Web Site, 60
 - Recent Projects, 61
 - New Web Site, 17
- commercial database systems, 284-285
- CompareValidator Web control, 250, 261-265
 - comparing input, 265-267
 - ErrorMessage property, 263
 - GreaterThanEqual property, 262
 - Operator property, 262
 - ValueToCompare property, 263
- comparison operators, 111-112
 - control structures, 112
 - WHERE clause, 320
- comparison validation, 249
- Completed Site Map Contents listing (20.2), 476
- compound conditionals, 125
- Compute Monthly Cost button, financial calculator web page, adding to, 79-80
- ComputeCostPerMonth function, 139-141
- ComputeCostPerMonth Function listing (6.6), 139
- concatenation operator, 112-113
- conditional control structures, 123-125
 - Else clause, 126-128
- conditional statements, compound conditionals, 125
- configuration
 - Visual Basic objects, 151-152
 - website membership, 522-526
 - websites, SMTP settings, 532-533
- Configuration and Deployment forum, 592
- configuration files, 61
- Configure Data Source Wizard, 308
- Configure ListView option (ListView Web control), 443-447
- confirmation messages, displaying, GridView Web control, 388
- ConfirmPasswordLabelText property (CreateUserWizard), 536
- connection strings, 309, 588

ConnectionString property (SqlDataSource)

- ConnectionString property (SqlDataSource), 313**
- constructors, 150**
 - creating objects with, 150
 - naming conventions, 150
 - parameters, 150-151
- content (Visual Web Developer)**
 - adding to websites, 62-64
 - deleting from websites, 64-65
 - renaming, 64-65
- content pages, 496**
 - creating, 507-510
- ContentPlaceHolder Web control**
 - adding to master pages, 500-506
 - default master pages, 512
- Continue command (Debug menu), 96**
- ContinueButtonText property (CreateUserWizard), 536**
- control structures, 121-122, 142**
 - comparison operators, 112
 - conditional control structures, 123-125
 - Else clause, 126-128
 - Functions
 - calling, 143
 - versus subroutines, 142
 - infinite loops, 131
 - looping control structures, 128
 - Do...Loop loops, 130-131
 - For...Next loops, 128-130
 - modularizing control structures, 131
 - code redundancy, 135-136
 - event handlers, 141
 - functions, 131
 - passing parameters, 136-138
 - returning values, 138-141
 - subroutines, 131-135
 - subroutines
 - example, 132
 - ShowWelcomeMessage, 136
 - versus functions, 142
- controls. See Web controls**
- ControlToValidate property (RequiredFieldValidator), 253-255**
- copying drop-down lists, 243**
- Create Your Own Website, 300**
- CreateUserButtonText property (CreateUserWizard), 536**
- CreateUserIconUrl property (Login control), 543**
- CreateUserText property (Login control), 543**
- CreateUserUrl property (Login control), 543**
- CreateUserWizard Web control, 533-535**
 - Answer property, 536
 - AnswerLabelText property, 536
 - ConfirmPasswordLabelText property, 536
 - ContinueButtonText property, 536
 - CreateUserButtonText property, 536
 - customizing, 535-537
 - Email property, 536
 - EmailLabelText property, 536
 - EmailRequiredErrorMessage property, 537
 - MailDefinition property, 539
 - PasswordLabelText property, 536
 - PasswordRequiredErrorMessage property, 537
 - Question property, 536
 - QuestionLabelText property, 536
 - UserName property, 536
 - UserNameLabelText property, 536
 - UserNameRequiredErrorMessage property, 536
- credentials, user accounts, 522**
- Criteria pane (SQL query window), 315**
- CSS (cascading style sheets), 36**
- CSS Properties window (Visual Web Developer), 16**
- CssClass property (GridView), 341**
- CssClass property (SiteMapPath), 479**
- CType function (Visual Basic), 117**
- current hour, determining, 123**
- current nodes (site maps), 479**

- CurrentNodeStyle property (SiteMapPath), 480**
 - CurrentTime Literal Control listing (8.1), 162**
 - custom templates, FormView Web control, 460-462**
 - customization**
 - CreateUserWizard, 535-537
 - DetailsView Web control
 - appearance, 350-351
 - paging interface, 349-350
 - Login Web control, 542
 - user accounts, 550
 - Visual Web Developer, 65-68
 - customizing appearance**
 - CheckBoxList Web control, 410-411
 - GridView Web control, 339-347, 356
 - levels, 339
 - RadioButtonList Web control, 410-411
 - SiteMapPath Web control, 478-481
 - TreeView Web control, 483-486
 - CustomValidator Web control, 274**
- D**
- data binding, 429, 436**
 - expressions, editing, 383
 - one-way data binding, 430
 - two-way data binding, 430
 - Web controls**
 - declarative markup, 435-436
 - specifying for, 430-434
 - WHERE filter expression, wildcards, 426-429, 437
 - data connections, SqlDataSource Web control, 308-310**
 - data source Web controls, 305-306**
 - AccessDataSource, 307
 - SqlDataSource, 307, 329-330
 - adding to web pages, 307-313
 - choosing data connections, 308-310
 - configuring SELECT statement, 310-312
 - ConnectionString property, 313
 - filtering data, 323-326
 - ID property, 313
 - markup, 313-314, 328-329
 - SelectCommand property, 313
 - sorting data, 326-327
 - testing queries, 312-313, 327-328
 - data sources, GridView Web control, changing, 346**
 - data template Web controls, ListView Web control, LayoutTemplate, 451**
 - data types, 104**
 - Boolean, 108
 - casting, 116-117
 - char, 288
 - Date Time, 108
 - Decimal, 108
 - Double, 107
 - integer types, 106-107
 - nchar, 288
 - nonintegral numeric types, 107-108
 - nvarchar, 288
 - Object, 109
 - string, 108
 - strings, 288
 - validation, 249
 - varchar, 288
 - variables, Visual Basic, 106-109
 - Visual Basic rules, 116-117
 - data Web controls, 333-337**
 - DetailsView control, 347-349
 - customizing appearance, 350-351
 - customizing paging interface, 349-350
 - GridView control, 337-338
 - AutoGenerateColumns property, 339
 - customizing appearance, 339-347, 356
 - DataKeyNames property, 339
 - DataSourceID property, 339

data Web controls

- declarative markup, 338-339
- online resources, 357
- paging, 351-353
- sorting, 351-355
- templated data Web controls, 441-442
 - FormView, 456-462
 - ListView, 442-456
- data-modification SQL statements, 363**
 - DELETE, 365
 - INSERT, 364-365
 - UPDATE, 365-366
- Database Design for Mere Mortals, 298*
- Database Publishing Wizard, 584-588**
- databases, 284, 302**
 - ASPNETDB, 524
 - ASPNETDB.MDF, 525
 - commercial database systems, 284-285
 - creating, 290-292
 - data source Web controls, 305-306
 - AccessDataSource, 307
 - SqlDataSource, 307-314, 323-330
 - data-modification SQL statements, 363
 - DELETE, 365
 - INSERT, 364-365
 - UPDATE, 365-366
 - deleting data
 - GridView Web control, 367-371
 - SqlDataSource Web control, 360-361
 - designing, 295-298
 - drop-down lists, populating, 243
 - inserting data
 - DetailsView Web control, 385-387
 - SqlDataSource Web control, 360-361
 - records
 - displaying, 456-458
 - paging, 458-459
 - replicating, 576, 583-584
 - Database Publishing Wizard, 584-588
 - output scripts, 586-588
 - SQL (Structured Query Language), 284, 305, 314
 - ORDER BY clause, 322-323
 - SELECT statement, 314-323
 - syntax, 314-315
 - viewing results of, 315-319
 - WHERE clause, 319-322
 - tables, 285-286
 - adding data to, 298-301
 - columns, 285-290, 302
 - creating, 292-298
 - modeling entities, 297-298
 - records, 285, 295-297
 - retrieving data, 330
 - updating data
 - GridView Web control, 373-384
 - SqlDataSource Web control, 360-361
- DataBindings dialog box, 383**
- DataFormatString property (GridView), 345**
- DataKeyNames property (GridView), 339, 373**
- DataSourceID property (GridView), 339**
- DateTime data type, 108**
- datetime column, 287**
- DateTime.Now.Hour, 123**
- DB2, 285**
- Debug menu commands, 96**
- debugger, 93**
 - breakpoints, setting, 94
 - Locals window, 96
 - running, 94
 - setting breakpoints, 94-96
 - stopping, 96
 - Watch window, 96
- debugging source code, financial calculator web pages, 93-96**
- Debugging Not Enabled dialog box, 94**
- Decimal data type, 108**
- declarative markup**
 - GridView Web control, 338-339
 - master pages, 501

DisplayAfter property (UpdateProgress)

- SqlDataSource Web control, 361-363
 - Web controls, data binding, 435-436
- declaring variables, Visual Basic, 103-105**
- declaratively setting properties, web controls, 52**
- Default Content of the Site Map listing (20.1), 473**
- Default Declarative Markup for a Master Page listing (21.1), 501**
- default events, 97**
- DELETE statement (SQL), 365**
- deleting columns, 302**
- deleting data**
 - DELETE statement, 365
 - GridView Web control, 367-371
 - SqlDataSource Web control, 360-361
- deployment (websites), 573, 592-593**
 - database replication, 583-588
 - process, 576-590
 - remote websites, 590-591
 - updating web.config settings, 588-590
 - uploading website files, 581-583
 - web-hosting companies, 574-576
- design requirements, web pages, 73-76**
- Design view (Visual Web Developer), 29-31**
 - editing HTML content, 33-40
- designing**
 - databases, 295-298
 - sitewide page templates, 502-506
 - web pages, requirements, 73-76
- Designing Active Server Pages, 246**
- DestinationPageUrl property (Login control), 543**
- DetailsView Web control, 347-349**
 - customizing appearance, 350-351
 - customizing Insertable DetailsView, 385-387
 - customizing paging interface, 349-350
 - displaying bit columns, 418-420
 - displaying hyperlinks, 420-425, 436
 - displaying images, 425-426
 - fields, 416-425, 436
 - BoundField, 417
 - ButtonField, 417
 - CheckBoxField, 417-420
 - CommandField, 417
 - HyperLinkField, 417, 420-425, 436
 - ImageField, 417, 425-426
 - TemplateField, 417, 437
 - inserting data, 385-386
- Development Web Server, hosting web pages, 12-13**
- Diagram pane (SQL query window), 315**
- dialog boxes**
 - Add New Item, 62-63, 290, 472
 - Add WHERE Clause, 324-326
 - Advanced SQL Generation Options, 360
 - AutoFormat, 346-347, 480-481
 - Choose Location, 59-60
 - DataBindings, 383
 - Debugging Not Enabled, 94
 - Insert Table, 504
 - ListItem Collection Editor, 227-228
 - Modify Style, 504-505
 - More Colors, 166-167
 - New Web Site, 17, 30-31, 58
 - New Web Site dialog box, 58
 - Open Web Site, 60
 - Options (Visual Web Developer), 66-67
 - Regular Expression Editor, 271
 - Select a Master Page, 507
- Different Message Is Displayed Based on the Current Hour listing (6.1), 124-126**
- Dim statement, 105, 109**
- Display property, validation Web control, 272, 276**
- DisplayAfter property (UpdateProgress), 567**

displaying confirmation messages, GridView Web control

- displaying confirmation messages, GridView Web control, 388
- displaying data
 - data Web controls, 333-337
 - DetailsView control, 347-351
 - GridView control, 337-347, 351-357
 - FormView Web control, 456-458
 - ListView Web control, 442-443
 - adding templates, 445-450
 - Configure ListView option, 443-447
 - ItemTemplate, 443-445
- displaying progress messages, UpdateProgress Web control, 566-567
- displaying site structure
 - Menu Web control, 487-491
 - TreeView Web control, 481-490
- displaying text
 - Label Web control, 163-164
 - BackColor property, 166-167
 - border properties, 168-170
 - Font property, 163-165, 169-171
 - ForeColor property, 166-167
 - formatting properties, 165-171
 - Height property, 170
 - setting properties
 - programmatically, 173
 - Text property, 164
 - Tooltip property, 170-171
 - Visible property, 170
 - Width property, 170
 - Literal Web control, 157-163
 - ID property, 159
 - Text property, 159-163
 - DisplayRememberMe property (Login control), 543
 - “Dissecting the Validation Controls,” 276
 - division (/) operator, 110
 - Do...Loop loops, 130-131
 - Do loops, 130-131
 - domain names, 574
 - Double data type, 107
 - downloading Microsoft SQL Server Management Studio Express Edition, 586
 - drop-down lists
 - adding items to, 224-225
 - collecting user input, 224
 - copying, 243
 - populating from databases, 243
 - properties, 232
 - rearranging order of items, 230
 - web pages, adding to, 225-232
 - DropDownList Web control, 224
 - adding items to, 224-225
 - AutoPostBack property, 406
 - BackColor property, 232
 - Choose a Genre option, 412
 - copying, 243
 - filtering results, 401-406
 - Font property, 232
 - ForeColor property, 232
 - listing genres, 402-404
 - populating from databases, 243
 - rearranging order of items, 230
 - Selected property, 228
 - SelectedItem property, 231
 - Text property, 229
 - Value property, 229
 - web pages, adding to, 225-232
 - DropDownList.aspx page, 230-231
 - Dynamic Help (Visual Web Developer), 69-70
 - dynamic web pages, 7-9, 61
 - definition, 61
 - serving, 10-12
 - static web pages, compared, 22-23
 - dynamically populating list Web controls, 397-398
 - DynamicEnableDefaultPopOutImage property (Menu control), 489
 - DynamicHorizontalOffset property (Menu control), 490
 - DynamicItemFormatString property (Menu control), 490
 - DynamicPopOutImageTextFormatString property (Menu Web control), 490

- DynamicPopOutImageUrl property (Menu control), 490
 - DynamicVerticalOffset property (Menu control), 490
- E**
- Edit field (GridView), 373
 - Edit Templates smart tag (FormView), 457
 - editable GridView, creating, 371-373
 - editing
 - data binding expressions, 383
 - HTML content, Designer, 33-40
 - editing data, UPDATE statement, 365-366
 - editing interface (GridView), customizing, 373-384
 - EditItemTemplate (FormView Web control), 460
 - EditItemTemplate (ListView Web control), 446
 - EditRowStyle property (GridView), 342
 - elements
 - form, 182-187
 - input, 180-182
 - siteMapNode, 473-476
 - elements (HTML)
 - block elements, 28
 - bold tag, 26
 - closing tag, 26
 - end tag, 26
 - form tag, 32
 - heading tag, 32
 - horizontal tag, 27
 - inline elements, 28
 - line break tag, 27
 - nested tags, 28-29
 - start tag, 26
 - table, 33-35, 37
 - when to use, 53
 - Else clause, conditional statements, 126-128
 - email
 - new user accounts, sending to, 538-540
 - sending from ASP.NET pages, 155
 - Email property (CreateUserWizard), 536
 - EmailLabelText property (CreateUserWizard), 536
 - EmailRequiredErrorMessage property (CreateUserWizard), 537
 - empty strings, 132
 - Empty Web Site template, Visual Web Developer, 17
 - EmptyDataRowStyle property (GridView), 342
 - EmptyDataTemplate (FormView Web control), 460
 - EmptyDataTemplate (ListView Web control), 447
 - EmptyDataText property (GridView), 341
 - Enabled property (drop-down list items), 228
 - end tag (HTML), 26
 - entities, tables, modeling, 297-298
 - enumerating items, list Web controls, 407-409
 - equal (=) operator, 112-113
 - ErrorMessage property (CompareValidator), 263
 - ErrorMessage property, validation Web control, 276
 - event handlers, 48-49, 141, 153
 - creating, 149
 - master pages, 514
 - modularizing control structures, 141
 - naming, 141
 - object events, creating, 153-154
 - Page_Load, 141
 - SelectedIndexChanged, 398-400
 - event handling text boxes, 203-205, 217
 - event-driven programming, 46
 - events, 45
 - default events, 97
 - objects, event handler creation, 153-154
 - RowDeleted, 370
 - RowDeleting, 370
 - “Examining ASP.NET’s Membership, Roles, and Profile,” 547, 550
 - “Examining ASP.NET’s Site Navigation,” 492
 - Execute SQL icon, 318
 - executed events, 46
 - executing code, 10

exercises

exercises

- Hour 4, 99-100
- Hour 5, 119
- Hour 6, 145
- Hour 8, 175
- Hour 10, 219-220
- Hour 11, 246
- Hour 12, 278-279
- Hour 13, 303-304
- Hour 14, 332
- Hour 15, 358
- Hour 16, 389-392
- Hour 17, 413-414
- Hour 18, 439
- Hour 19, 466
- Hour 20, 493-494
- Hour 21, 519
- Hour 22, 552-553
- Hour 23, 570-571
- Visual Web Developer, 54-55, 72

ExpandImageUrl property (TreeView), 485

ExpandTooltip property (TreeView), 485

explicit casting, data types, 116

expressions

- regular expressions, 270
- WHERE, wildcards, 426-429, 437

eXtensible Markup Language (XML). See XML (eXtensible Markup Language)

extraneous whitespace (HTML), 27-28

F

false conditional statements, 126-128

Field level (GridView Web control), 340

fields. See also columns

- DetailsView Web control, 416-425, 436
 - BoundField, 417
 - ButtonField, 417
 - CheckBoxField, 417-420
 - CommandField, 417
 - HyperLinkField, 417, 420-425, 436
 - ImageField, 417, 425-426
 - TemplateField, 417, 437

- GridView Web control, 416-425, 436
 - BoundField, 417
 - ButtonField, 417
 - CheckBoxField, 417-420
 - CommandField, 417
 - editing, 377-378
 - formatting, 344-346
 - HyperLinkField, 417, 420-425, 436
 - ImageField, 417, 425-426
 - marking as Read-Only, 375-377
 - TemplateField, 417, 437

Fight Club, 301

File menu commands

- New Web Site, 17, 58
- Open Web Site, 60
- Recent Projects, 61

file systems, hosting websites, 59

file types, websites, 61-62

files

- adding to websites, 62-64
- configuration files, 61
- removing from websites, 64-65
- renaming, 64-65
- script files, 61
- style sheet files, 61
- Web.sitemap, 473
- XML files, 307, 470

filter expressions, WHERE, wildcards, 426-429, 437

filtering data

- CheckBoxList Web control, 412
- SqlDataSource Web control, 323-326

filtering results, DropDownList Web control, 401-406

financial calculator web page

- Compute Monthly Cost button, 79-80
- Label Web control, 81-82
- source code, 85-86
 - debugging, 93-96
 - examining, 92-93
 - viewing, 87-91
- testing, 86-92
- TextBox Web controls, reading values in, 84-85
- UI (user interface)
 - creating, 76-82
 - design requirements, 74-76

GridView Web control

- TextBox Web controls, 77-79
 - writing source code for, 82-86
 - validation Web controls, 272-274
 - reducing code redundancy, 135-136
 - returning values, 138-141
 - subroutines, compared, 142
 - syntax, 139
 - formatting rules (XML), 475
 - forms. *See* Web Forms
 - forms-based authentication, 523
 - FormView data Web control
 - AllowPaging property, 458-459
 - custom templates, 460-462
 - EditItemTemplate, 460
 - EmptyDataTemplate, 460
 - FooterTemplate, 460
 - HeaderTemplate, 460
 - InsertItemTemplate, 460
 - ItemTemplate, 460
 - PagerSettings, 458
 - PagerTemplate, 460
 - paging data, 456-458
 - paging records, 458-459
 - smart tag, 456
 - templates, 459-460
 - Friend keyword, 143
 - From property (MailDefinition), 539
 - functionality, master pages, testing, 515-516
 - functions, 152
 - calling, 143
 - ComputeCostPerMonth, 139-141
 - CType, 117
 - modularizing control structures, 131
 - passing parameters to, 136-138
 - GridView Web control, 337-338, 366
 - allowing users to delete data, 367-371
 - AllowPaging property, 349, 352
 - AlternatingRowStyle property, 342
 - AutoGenerateColumns property, 339
 - BackColor property, 341
- G**
- general nodes (site maps), 479
 - Genre TemplateField's Declarative Markup listing (18.1), 435
 - genres, listing, DropDownList Web control, 402-404
 - Google Gmail, 557
 - greater than (>) operator, 112
 - greater than or equal (>=) operator, 112
 - GreaterThanEqual property (CompareValidator), 262
 - GridLines property (GridView), 341
 - "GridView Examples for ASP.NET", 357

GridView Web control

- BackColorUrl property, 341
 - blank strings, 378-379
 - BorderColor property, 341
 - BorderStyle property, 341
 - BorderWidth property, 341
 - changing data source, 346
 - creating editable GridViews, 371-373
 - CssClass property, 341
 - customizing appearance, 339-347, 356
 - levels, 339
 - customizing editing interface, 373-384
 - DataFormatString property, 345
 - DataKeyNames property, 339, 373
 - DataSourceID property, 339
 - declarative markup, 338-339
 - displaying bit columns, 418-420
 - displaying confirmation messages, 388
 - displaying hyperlinks, 420-425, 436
 - displaying images, 425-426
 - Edit field, 373
 - editing fields, 377-378
 - EditRowStyle property, 342
 - EmptyDataRowStyle property, 342
 - EmptyDataText property, 341
 - fields, 416-425, 436
 - BoundField, 417
 - ButtonField, 417
 - CheckBoxField, 417-420
 - CommandField, 417
 - HyperLinkField, 417, 420-425, 436
 - ImageField, 417, 425-426
 - TemplateField, 417, 437
 - Font property, 341
 - FooterStyle property, 342
 - ForeColor property, 341
 - GridLines property, 341
 - HeaderStyle property, 342, 345
 - HeaderText property, 345
 - ItemStyle property, 345
 - ListView Web control, compared, 464
 - marking fields as Read-Only, 375-377
 - nulls, 378-379
 - online resources, 357
 - PagerSettings property, 352
 - PagerStyle property, 342
 - PageSize property, 352
 - PageView property, 352
 - paging, 351-353
 - RowStyle property, 342
 - SelectedRowStyle property, 342
 - ShowFooter property, 341
 - ShowHeader property, 341
 - ShowHeaderWhenEmpty property, 341
 - SortedAscendingCellStyle property, 343
 - SortedAscendingHeaderStyle property, 343
 - SortedDescendingCellStyle property, 343
 - SortedDescendingHeaderStyle property, 343
 - sorting, 351-355
 - updating rules, 373-384
- GridView's Markup listing (15.1), 338**
- GroupName property (RadioButton), 235-237**
- GroupTemplate (ListView Web control), 447, 463**
- ## H
- Handles keyword, 154**
 - head ContentPlaceHolder Web control, 512**
 - Head First SQL, 298**
 - header region, sitewide page templates, creating, 504-505**
 - HeaderRegion ContentPlaceHolder Web control, 512**
 - HeaderStyle property (GridView), 342, 345**
 - HeaderTemplate (FormView Web control), 460**
 - HeaderText property (GridView), 345**
 - heading tag (HTML), 32**
 - Height property (Label), 170**
 - Help, Visual Web Developer, 69-70**
 - Hernandez, Michael, 298**

- horizontal tag (HTML), 27
 - hosting websites, 59
 - Development Web Server, 12-13
 - web-hosting companies, 70, 574-576
 - Hour 1 quiz, 23-24
 - Hour 2 exercises, 54-55
 - Hour 2 quiz, 53-54
 - Hour 3 quiz, 71-72
 - Hour 4 exercises, 99-100
 - Hour 4 quiz, 98-99
 - Hour 5 exercises, 119
 - Hour 5 quiz, 118-119
 - Hour 6 exercises, 145
 - Hour 6 quiz, 144
 - Hour 7 quiz, 155-156
 - Hour 8 exercises, 175
 - Hour 8 quiz, 173-174
 - Hour 9 quiz, 199-200
 - Hour 10 exercise, 219-220
 - Hour 10 quiz, 218-219
 - Hour 11 exercises, 246
 - Hour 11 quiz, 244-245
 - Hour 12 exercises, 278-279
 - Hour 12 quiz, 277-278
 - Hour 13 exercises, 303-304
 - Hour 13 quiz, 303
 - Hour 14 exercises, 332
 - Hour 14 quiz, 331-332
 - Hour 15 exercises, 358
 - Hour 15 quiz, 357-358
 - Hour 16 exercises, 389-392
 - Hour 16 quiz, 388-389
 - Hour 17 exercises, 413-414
 - Hour 17 quiz, 412-413
 - Hour 18 exercise, 439
 - Hour 18 quiz, 438
 - Hour 19 exercises, 466
 - Hour 19 quiz, 464-465
 - Hour 20 exercises, 493-494
 - Hour 20 quiz, 493
 - Hour 21 exercises, 519
 - Hour 21 quiz, 517-518
 - Hour 22 exercises, 552-553
 - Hour 22 quiz, 551-552
 - Hour 23 exercises, 570-571
 - Hour 23 quiz, 569-570
 - Hour 24 quiz, 593-594
 - HoverNodeStyle property (TreeView), 484
 - HTML, 25-29
 - colors, 167
 - content, examining, 40-43
 - CSS (cascading style sheets), 36
 - editing, Visual Web Developer, 33-40
 - elements
 - block elements, 28
 - bold tag, 26
 - closing tag, 26
 - end tag, 26
 - form, 182-187
 - form tag, 32
 - heading tag, 32
 - horizontal tag, 27
 - inline elements, 28
 - input, 180-182
 - line break tag, 27
 - nested tags, 28-29
 - start tag, 26
 - table, 33-35, 37
 - when to use, 53
 - extraneous whitespace, 27-28
 - web pages, creating, 29-43
 - HTML Markup Received by the Browser After the User Has Clicked the Button listing, 89
 - HTML Markup Received by the Browser When First Visiting the Page listing (4.2), 87
 - HTML Received by the Browser listing (9.5), 193
 - HTML Received by the Browser When Visiting Default.aspx listing (2.3), 42
 - HTML-Formatted Email listing (22.2), 538
 - HyperLinkField, 417, 420-425, 436
 - hyperlinks, displaying, 420-425, 436
- I**
- IBM DB2, 285
 - IBM Informix, 285
 - ID property (Literal Web control), 159
 - ID property (SqlDataSource), 313
 - If statements, 123-125
 - Else clause, 126-128

IIS (Internet Information Server)

IIS (Internet Information Server),
22-23

hosting websites, 59

image files, websites, 61

Image Web control, web pages,
adding to, 38-39

ImageField, 417, 425-426

images

displaying, 425-426

loading, Ajax, 568

ImageSet property

(TreeView), 485

immutable values, 104

immutable values, variables, 104

implicit casting, data types, 116

improperly nested tags
(HTML), 29

inactive user accounts,
creating, 540

inequality (<>) operator, 112

infinite loops, 131

Informix, 285

inheriting master pages, 509-510

inline elements (HTML), 28

input. *See* User input

INSERT statement (SQL),
364-365

Insert Table dialog box, 504

Insertable DetailsView,
customizing, 385-387

inserting data

DetailsView Web control,
385-386

INSERT statement, 364-365

SqlDataSource Web control,
360-361

InsertItemTemplate
(FormView), 460

InsertItemTemplate
(ListView), 447

installation

.NET Framework, 14-16

SQL Server 2008, 14-16

Visual Web Developer, 14-16

instances, objects, 148

instantiation

definition, 84

objects, 149

instructions, computers, 102

Int column, 287

integer types, Visual Basic
variables, 106-107

IntelliSense, 51

intranets, 523

ISBNs, 421

IsBodyHtml property
(MailDefinition), 539

Italic subproperty (Label control
Font property), 164

Items property

CheckBoxList Web control,
407-409

RadioButtonList Web control,
407-409

ItemSeparatorTemplate (ListView
Web control), 447

ItemStyle property
(GridView), 345

ItemTemplate (FormView Web
control), 460

ItemTemplate (ListView Web
control), 443-445

ItemTemplate (ListView Web
control), 447

ItemWrap property (Menu
control), 490

J-K

JavaScript, 557-558

keywords

access modifiers, 143

ByRef, 137

ByVal, 137

Friend, 143

Handles, 154

ORDER BY, SELECT
statement, 322-323

Private, 143

Protected, 143

Protected Friend, 143

Public, 143

subroutine parameters, 137

WHERE clause, SELECT
statement, 319-322

L

Label Web control, 163

borders, 168

displaying text, 163-164

BackColor property,
166-167

border properties,
168-170

- Font property, 163-165, 169-171
- ForeColor property, 166-167
- formatting properties, 165-171
- Height property, 170
- setting properties programmatically, 173
- Text property, 164
- Tooltip property, 170-171
- Visible property, 170
- Width property, 170
- financial calculator web page, adding to, 81-82
- Literal Web control, compared, 172-173
- languages (programming), 102-103**
- LayoutTemplate (ListView Web control), 447, 451**
- leaf nodes (TreeView), 483**
- LeafNodeStyle property (TreeView), 484**
- less than (<) operator, 112**
- less than or equal (<=) operator, 112**
- levels, GridView Web control customization, 339**
- LevelStyles property (TreeView), 484**
- limiting characters, text boxes, 213-214**
- line break tag (HTML), 27**
- list items (drop-down lists), 224**
 - rearranging order of, 230
- list Web controls, 394-395**
 - AutoPostBack property, 401
 - binding data to, 395-397
 - CheckBoxList, 406-407
 - customizing appearance, 410-411
 - Items property, 407-409
 - SelectedItem property, 408-410
 - SelectedValue property, 408-410
 - DropDownList
 - Choose a Genre option, 412
 - filtering results, 401-406
 - listing genres, 402-404
 - dynamically populating, 397-398
 - programmatically responding to changed selections, 398-401
 - RadioButtonList, 406-407
 - customizing appearance, 410-411
 - filtering data, 412
 - Items property, 407-409
 - SelectedItem property, 408-410
 - SelectedValue property, 408-410
- listing genres, DropDownList Web control, 402-404**
- listings**
 - 2.1 (Snippet of HTML with Extraneous Whitespace), 27
 - 2.2 (Markup Generated by the Designer), 40
 - 2.3 (HTML Received by the Browser When Visiting Default.aspx), 42
 - 2.4 (Page_Load Event Handler Fires in Response to the Page's Load Event), 49
 - 4.1 (PerformCalcButton Button's Click Event Handler), 85-86
 - 4.2 (HTML Markup Received by the Browser When First Visiting the Page), 87
 - 4.3 (HTML Markup Received by the Browser After the User Has Clicked the button, 89
 - 6.1 (Different Message Is Displayed Based on the Current Hour), 124-126
 - 6.2 (Page_Load Event Handler), 132
 - 6.3 (Message Is Displayed Four Times when the Button Web Control Is Clicked), 133
 - 6.4 (Code to Display the "Welcome to my Website"), 135
 - 6.5 (ShowWelcomeMessage Subroutine Accepts a Parameter), 138
 - 6.6 (ComputeCostPerMonth Function), 139
 - 6.7 (PerformCalcButton_Click Event Handler Calls the ComputeMonthlyCost Function), 140
 - 8.1 (CurrentTime Literal Control), 162

listings

- 9.1 (First Draft of the BMI Calculator), 181
 - 9.2 (Second Draft of the BMI Calculator), 182
 - 9.3 (<form> Element has been Added), 184
 - 9.4 (Browser Receives Elements), 191
 - 9.5 (HTML Received by the Browser), 193
 - 9.6 (Code for the BMI Calculation), 195
 - 10.1 (Click Event Handler), 205
 - 11.1 (Code for the Button's Click Event Handler), 237
 - 11.2 (User's Ice Cream Preferences Are Displayed), 241
 - 15.1 (GridView's Markup), 338
 - 16.1 (SqlDataSource Control's Markup), 362
 - 16.2 (TemplateField Has an ItemTemplate), 381
 - 18.1 (Genre TemplateField's Declarative Markup), 435
 - 19.1 (Simple ListView Control's Declarative Markup), 444
 - 19.2 (ListView Uses Two Templates), 449
 - 20.1 (Default Content of the Site Map), 473
 - 20.2 (Completed Site Map Contents), 476
 - 21.1 (Default Declarative Markup for a Master Page), 501
 - 21.2 (Markup of a Content Page), 508
 - 21.3 (Master Page Has Two Event Handlers), 514
 - 22.1 (Plain-Text Email Includes the User's Username), 538
 - 22.2 (HTML-Formatted Email), 538
 - 23.1 (Page Includes Two Labels), 561
- ListItem Collection Editor dialog box, 227-228**
- lists (drop-down)**
- properties, 232
 - rearranging order of items, 230
- ListView data Web control, 442-443**
- adding templates, 445-450
 - AlternatingItemTemplate, 446
 - CommandArgument property, 452
 - CommandName property, 452
 - Configure ListView option, 443, 445-447
 - EditItemTemplate, 446
 - EmptyDataTemplate, 447
 - GridView Web control, compared, 464
 - GroupTemplate, 447, 463
 - InsertItemTemplate, 447
 - ItemSeparatorTemplate, 447
 - ItemTemplate, 443-445, 447
 - LayoutTemplate, 447, 451
 - pagination data, 450, 453-456
 - sorting data, 450-452
- ListView Uses Two Templates listing (19.2), 449**
- Literal Web control**
- displaying text, 157-163
 - ID property, 159
 - Text property, 159-163
 - Label Web control, compared, 172-173
- LiteralTime.aspx web page, 161-162**
- loading images, Ajax, 568**
- Locals window (debugger), 96**
- Locations, websites, Visual Web Developer, 58-60**
- logging in**
- user accounts, 541-542
 - login page URL specification, 544-545
- logging out user accounts, 543-544**
- logical operators, WHERE clause, 320**
- login page URLs, specifying, 544-545**
- Login Web control, 541-545**
- customizing, 542
 - DestinationPageUrl property, 543
 - DisplayRememberMe property, 543
 - Orientation property, 543
 - RememberMeSet property, 543
 - TextLayout property, 543
 - VisibleWhenLoggedIn property, 543

logins, Remember Me Next Time option, 541

Long data type, 107

loops, 128

Do loops, 130

bodies, 131

Do...Loop loops, 130-131

For...Next loops, 128-130

infinite loops, 131

M

MailDefinition property (CreateUserWizard), 539

MailMessage class, 155

main region, sitewide page templates, creating, 506

Manage Styles window (Visual Web Developer), 16-17

managing users, 526-528

maps, site maps, 307

markup

GridView Web control, 338-339

master pages, 501

SqlDataSource Web control, 313-314, 328-329, 361-363

Web controls, data binding, 435-436

Markup Generated by the Designer listing (2.2), 40

Markup of a Content Page listing (21.2), 508

masked text boxes, 216

Master Page Has Two Event Handlers listing (21.3), 514

master pages, 63, 492, 495-499, 516, 544

creating, 500-506

declarative markup, 501

event handlers, 514

inheriting, 509-510

nested master pages, 513, 517

providing default content in, 510-512

source code, 513-516

testing, 515-516

MasterPageFile attribute, 509

math operators, 110-111

MaxLength property (TextBox), 214

membership, configuring websites to support, 522-526

Menu Web control, 470

displaying site structure, 487-489, 491

DynamicEnableDefaultPopOutImage property, 489

DynamicHorizontalOffset property, 490

DynamicItemFormatString property, 490

DynamicPopOutImageTextFormatString property, 490

DynamicPopOutImageUrl property, 490

DynamicVerticalOffset property, 490

ItemWrap property, 490

Orientation property, 490

StaticEnableDefaultPopOutImage property, 489

StaticItemFormatString property, 490

StaticPopOutImageTextFormatString property, 490

StaticPopOutImageUrl property, 490

StaticSubMenuIndent property, 490

menus, showing site structures, 487-488

Message Is Displayed Four Times when the Button Web Control Is Clicked listing (6.3), 133

methods, 45, 150

calling objects, 149, 152-153
constructors, 150

creating objects with, 150

naming conventions, 150

parameters, 150-151

Microsoft IIS (Internet Information Server), 22-23

Microsoft SQL Server 2008 Express Edition. See SQL Server 2008 Express Edition

Microsoft SQL Server Management Studio Express Edition, downloading, 586

minimum requirements, web-hosting companies, 575

Mitchell, Scott, 300

Modify Style dialog box, 504-505

modularizing control structures, 131

code redundancy, 135-136

event handlers, 141

modularizing control structures

- functions, 131
 - passing parameters to, 136-138
 - returning values with, 138-141
 - versus subroutines, 142
- subroutines, 131-135
 - example, 132
 - passing parameters to, 136-138
 - ShowWelcomeMessage, 136
 - versus functions, 142
- money column, 287
- More Colors dialog box, 166-167
- mortgage calculator web page. *See* financial calculator web page
- moving
 - Visual Web Developer windows, 68-69
 - websites, 71
- multiline text boxes, 217
 - creating, 205-208
- multiplication (*) operator, 110
- mutable values, variables, 104
- mutually exclusive radio buttons, 233
- MySQL, 285

N

- Name subproperty (Label control Font property), 164
- naming
 - event handlers, 141
 - variables, Visual Basic, 103-106

- naming conventions, constructors, 150
- navigation region, sitewide page templates, creating, 505-506
- navigation Web controls, 470
- NBA (National Basketball Association), 22-23
- nchar data type, 288
- nested master pages, 513, 517
- nested tags (HTML), 28-29
- .NET Framework, 7
 - installing, 14-16
- New Web Site command (File menu), 17, 58
- New Web Site dialog box, 17, 30-31, 58
- NodeIndent property (TreeView), 485
- nodes
 - SiteMapPath Web control, 479
 - TreeView Web control, 483
- NodeStyle property (SiteMapPath), 480
- NodeStyle property (TreeView), 484
- NodeWrap property (TreeView), 486
- NoExpandImageUrl property (TreeView), 485
- nonintegral numeric types, 107-108
- not equal (<>) operator, 112
- nulls, GridView Web control, 378-379
- numbers, integer types, 106-107
- nvarchar data type, 288
- nvarchar(n) column, 287

O

- Object data type, 109
- object-oriented programming (OOP), 45
 - event-driven programming, 46
 - executed events, 46
 - fired events, 46
 - raised events, 46
 - out-of-order execution, 46-48
- objects, 45, 147-148, 154-155
 - calling methods, 149
 - classes, 148-149
 - common tasks, 149
 - creating, 150-151
 - definition, 148
 - event handlers, creating, 149, 153-154
 - instances, 148
 - instantiation, 149
 - definition, 84
 - methods, calling, 152-153
 - properties, setting, 151-152
 - role of, 149
 - setting property values, 149
- one-way data binding, 430
- online resources
 - ASP.NET, 71
 - GridView Web control, 357
- OOP (object-oriented programming), 45
 - event-driven programming, 46
 - executed events, 46
 - fired events, 46
 - raised events, 46
 - out-of-order execution, 46-48

PathDirection property (SiteMapPath)

- Open Web Site command (File menu), 60
 - Open Web Site dialog box, 60
 - OpenDesigns.org, 506
 - opening websites, Visual Web Developer, 60-61
 - Operator property (CompareValidator), 262
 - operators, 110
 - = operator, 104
 - arithmetic operators, 110-111
 - assignment operators, 113-115
 - shorthand, 114-115, 118
 - binary, 110
 - comparison operators, 111-112
 - control structures, 112
 - WHERE clause, 320
 - concatenation operator, 112-113
 - logical operators, WHERE clause, 320
 - precedence, 111
 - unary, 111
 - Options dialog box (Visual Web Developer), 66-67
 - Oracle, 284
 - ORDER BY clause, SELECT statement, 322-323
 - Orientation property (Login control), 543
 - Orientation property (Menu control), 490
 - out-of-order execution, OOP (object-oriented programming), 46-48
 - output scripts, replicating databases, 586-588
 - Overline subproperty (Label control Font property), 164
 - overloaded classes, 151
- P**
- Page Includes Two Labels listing (23.1), 561
 - page numbers, 347
 - PageIndex property (GridView), 352
 - pager rows, 347
 - PagerSettings property (DetailsView), 349
 - (FormView), 458
 - (GridView), 352
 - PagerStyle property (GridView), 342
 - PagerTemplate (FormView), 460
 - PageSize property (GridView), 352
 - Page_Load event handler, 49, 132, 141
 - Page_Load Event Handler Fires in Response to the Page's Load Event listing (2.4), 49
 - Page_Load Event Handler listing (6.2), 132
 - paging GridView Web control, 351-353
 - paging data, ListView Web control, 450, 453-456
 - paging interface, DetailsView Web control, customizing, 349-350
 - paging records, FormView Web control, 458-459
 - Palahniuk, Chuck, 301
 - parameterized queries, 329
 - parameters
 - constructors, 150-151
 - functions, passing to, 136-138
 - SqlDataSource Web control, 362
 - subroutines
 - keywords, 137
 - passing to, 136-138
 - parent nodes (TreeView), 483
 - parentheses, compound conditionals, 125
 - ParentNodeStyle property (TreeView), 484
 - partial page postbacks, Ajax, 556-557
 - progress messages for, 566-567
 - "Passing Information Between Content and Master Pages," 517
 - Password property (TextBox), 208
 - password text boxes, 209
 - creating, 205, 208-211
 - PasswordLabelText property (CreateUserWizard), 536
 - PasswordRecovery Web control, 550
 - PasswordRequiredErrorMessage property (CreateUserWizard), 537
 - path separators, 479
 - PathDirection property (SiteMapPath), 480

PathSeparator property (SiteMapPath)

PathSeparator property (SiteMapPath), 480

PathSeparatorStyle property (SiteMapPath), 480

pattern validation, 250

PerformCalcButton Button's Click Event Handler listing (4.1), 85-86

PerformCalcButton_Click Event Handler Calls the ComputeMonthlyCost Function listing (6.7), 140

personal computer, hosting via, 576

Plain-Text Email Includes the User's Username listing (22.1), 538

populating databases, drop-down lists, 243

populating list Web controls, 397-398

ports, 12

postback forms, 192, 198

- list Web controls, 398-401
- redirect forms, compared, 185-187
- refreshing web pages, 198

PostgreSQL, 285

precedence, operators, 111

PrettyTextBox.aspx page, 215

primary key columns, 288-290

Private keyword, 143

processing user input, Web Forms, 192-194

programmatically setting properties, web controls, 52

programming languages, 102-103

- common concepts, 103
- websites, choosing, 60

progress messages, displaying, UpdatePanel Web control, 566-567

properly nested tags (HTML), 29

properties, 45. *See also names of specific properties*

- objects, setting, 151-152

Properties window, GridView Web control, formatting from, 340-343

property values, objects, setting, 149

Protected Friend keyword, 143

Protected keyword, 143

providers, site maps, 492

pseudocode, 48

Public keyword, 143

Q

queries (SQL)

- parameterized queries, 329
- results, viewing, 315-323
- testing, 312-313
 - WHERE clauses, 327-328

query window (SQL), 315

querystrings, 185

Question property (CreateUserWizard), 536

QuestionLabelText property (CreateUserWizard), 536

quizzes

Hour 1, 23-24

Hour 2, 53-54

Hour 3, 71-72

Hour 4, 98-99

Hour 5, 118-119

Hour 6, 144

Hour 7, 155-156

Hour 8, 173-174

Hour 9, 199-200

Hour 10, 218-219

Hour 11, 244-245

Hour 12, 277-278

Hour 13, 303

Hour 14, 331-332

Hour 15, 357-358

Hour 16, 388-389

Hour 17, 412-413

Hour 18, 438

Hour 19, 464-465

Hour 20, 493

Hour 21, 517-518

Hour 22, 551-552

Hour 23, 569-570

Hour 24, 593-594

R

radio buttons, web pages, 233-238, 244

- mutually exclusive radio buttons, 233

- RadioButton Web control, 233-238, 244
 - GroupName property, 235-237
 - Text property, 235-237
 - RadioButtonList Web control, 406-407
 - customizing appearance, 410-411
 - filtering data, 412
 - Items property, 407-409
 - SelectedItem property, 408-410
 - SelectedValue property, 408-410
 - raised events, 46
 - range input validation, 249
 - RangeValidator Web control, 250, 268-269
 - rearranging list items, 230
 - Recent Projects command (File menu), 61
 - records, 285
 - displaying, FormView Web control, 456-458
 - paging, FormView Web control, 458-459
 - uniquely identifying, 295-297
 - redirect forms, postback forms, compared, 185-187
 - refreshing web pages, postback forms, 198
 - Regular Expression Editor dialog box, 271
 - regular expressions, 270
 - RegularExpressionValidator Web control, 250, 269-271, 277
 - Remember Me Next Time option (logins), 541
 - RememberMeSet property (Login control), 543
 - renaming website content, 64-65
 - RenderCurrentNodeAsLink property (SiteMapPath), 480
 - rendered source code, financial calculator web page, viewing, 87-91
 - replicating databases, 576, 583-584
 - Database Publishing Wizard, 584-588
 - requests, web requests, 9
 - required field input validation, 249
 - RequiredFieldValidator Web control, 250-254
 - ControlToValidate property, 253-255
 - features, 261
 - specifying error messages, 255-256
 - specifying validation, 253-255
 - testing, 256-261
 - resizing
 - text boxes, 212-213
 - Visual Web Developer windows, 68
 - Results pane (SQL query window), 316
 - roles, users, classifying, 528-529
 - root nodes (site maps), 479
 - root nodes (TreeView), 483
 - RootNodeStyle property (SiteMapPath), 480
 - RootNodeStyle property (TreeView), 484
 - Row level (GridView Web control), 340
 - RowDeleted event, 370
 - RowDeleting event, 370
 - rows. *See* records
 - RowStyle property (GridView), 342
 - rules
 - access rules, 529-532
 - Visual Basic data types, 116-117
 - runtime errors, input, proper format, 92
- ## S
- Salinger, J.D., 301
 - Sams Teach Yourself Active Server Pages 3.0 in 21 Days*, 246
 - Sams Teach Yourself ASP.NET 2.0 in 24 Hours*, 246
 - Sams Teach Yourself ASP.NET 3.5 in 24 Hours*, 246
 - Sams Teach Yourself ASP.NET in 24 Hours*, 246
 - Sams Teach Yourself HTML and CSS in 24 Hours*, 26
 - Sams Teach Yourself SQL in 10 Minutes*, 330
 - ScriptManager Web control, 558, 560
 - script files, 61
 - scripts, client-side script, 61, 258

Second Draft of the BMI Calculator listing

- Second Draft of the BMI Calculator listing (9.2), 182**
- security, authentication**
 - displaying content based on, 545-547
 - forms-based authentication, 523
 - Windows authentication, 523
- Select a Master Page dialog box, 507**
- Select Master Page check box, 511**
- SELECT statement**
 - optional clauses, 315
 - SqlDataSource Web control, configuring, 310-313
 - WHERE filter expression, wildcards, 426-429, 437
- SELECT statement (SQL), 314**
 - ORDER BY clause, 322-323
 - syntax, 314-315
 - viewing results of, 315-319
 - WHERE clause, 319-322
 - testing queries, 327-328
- SelectCommand property (SqlDataSource), 313**
- selected node (TreeView), 483**
- Selected property (DropDownList), 228**
- SelectedIndexChanged event handler, 398-400**
- SelectedItem property**
 - CheckBoxList Web control, 408-410
 - RadioButtonList Web control, 408-410
- SelectedItem property (DropDownList), 231**
- SelectedNodeStyle property (TreeView), 484**
- SelectedRowStyle property (GridView), 342**
- SelectedValue property**
 - CheckBoxList Web control, 408-410
 - RadioButtonList Web control, 408-410
- Send method, 153**
- sending**
 - ASP.NET pages, 155
 - new user accounts, 538-540
- "Sending Email in ASP.NET," 550**
- sequential execution, source code, 46**
- sequential flow, 122**
- server-side code, 45, 91**
 - master pages, 513-516
- server-side validation, 258**
- servers**
 - SMTP servers, 532
 - web servers, 9
- serving web pages**
 - dynamic web pages, 10-12
 - static web pages, 9-10
- Short data type, 107**
- shorthand, assignment operators, 114-115, 118**
- ShowExpandCollapse property (TreeView), 486**
- ShowFooter property (GridView), 341**
- ShowHeader property (GridView), 341**
- ShowHeaderWhenEmpty property (GridView), 341**
- ShowLines property (TreeView), 486**
- ShowWelcomeMessage subroutine, 135-138**
- ShowWelcomeMessage Subroutine Accepts a Parameter listing (6.5), 138**
- sign-in pages, 541**
- Simmons, Bill, 301**
- Simple ListView Control's Declarative Markup listing (19.1), 444**
- Single numeric data type, 107**
- site maps, 307, 469, 491**
 - adding to projects, 472-474
 - breadcrumbs, displaying, 476-478
 - building, 474-476
 - customizing appearance, 478-481
 - providers, 492
 - site structure, displaying, 481-491
 - structural definition, 471-472
- site navigation, 469-471**
 - customizing appearance, 478-481
 - displaying breadcrumbs, SiteMapPath, 476-478
 - displaying site structure, 481-491
 - nodes, 479
 - path separators, 479

SQL (Structured Query Language)

- site structures, displaying**
 - Menu, 487-491
 - TreeView, 481-490
- siteMapNode element, 473-476**
- SiteMapPath Web control, 470**
 - BackColor property, 479
 - BorderColor property, 479
 - BorderStyle property, 479
 - BorderWidth property, 479
 - CssClass property, 479
 - CurrentNodeStyle property, 480
 - customizing appearance, 478-481
 - displaying breadcrumbs, 476-478
 - Font property, 479
 - nodes, 479
 - NodeStyle property, 480
 - path separators, 479
 - PathDirection property, 480
 - PathSeparator property, 480
 - PathSeparatorStyle property, 480
 - RenderCurrentNodeAsLink property, 480
 - RootNodeStyle property, 480
- sitewide page templates**
 - content pages, creating, 507-510
 - designing, 502-506
 - footer region, creating, 506
 - header region, creating, 504-505
 - main region, creating, 506
- master pages, 496-499
 - creating, 500-506
 - default content, 510-512
 - source code, 513-516
 - navigation region, creating, 505-506
- Size subproperty (Label control Font property), 165**
- sizing**
 - text boxes, 212-213
 - Visual Web Developer, 68
- smart tags, 308**
 - FormView Web control, 456
- SMTP servers, 532**
- SMTP settings, websites, configuring, 532-533**
- SmtplibClient class, 155**
- Snippet of HTML with Extraneous Whitespace listing (2.1), 27**
- Solution Explorer (Visual Web Developer), 16**
 - adding content to websites, 62-64
- SortedAscendingCellStyle property (GridView), 343**
- SortedAscendingHeaderStyle property (GridView), 343**
- SortedDescendingCellStyle property (GridView), 343**
- SortedDescendingHeaderStyle property (GridView), 343**
- sorting data**
 - GridView Web control, 351-355
 - ListView Web control, 450-452
 - SqlDataSource Web control, 326-327
- source code**
 - event handlers, 48-49
 - financial calculator web page, 85-86
 - viewing, 87-91
 - master pages, 513-516
 - OOP (object-oriented programming), 45
 - event-driven programming, 46
 - executed events, 46
 - fired events, 46
 - out-of-order execution, 46-48
 - raised events, 46
 - sequential execution, 46
 - server-side code, 45
 - web controls, programmatically working with, 49-52
 - web pages
 - debugging, 93-96
 - examining, 44-52, 92-93
 - writing for, 82-86
 - source code programming language, websites, choosing, 60**
 - Source view (Visual Web Developer), 29-31**
 - Split view (Visual Web Developer), 30-33**
 - SQL (Structured Query Language), 284, 305, 314**
 - multiple tables, retrieving data from, 330
 - queries

SQL (Structured Query Language)

- parameterized queries, 329
- testing, 312-313, 327-328
- SELECT statement, 314
 - configuring, 310-312
 - ORDER BY clause, 322-323
 - syntax, 314-315
 - viewing results of, 315-319
 - WHERE clause, 319-322
 - viewing query results, Visual Web Developer, 315-323
- SQL pane (SQL query window), 316**
- SQL Server 2008, 7-8, 283-285**
 - databases
 - Books table, 298-301
 - creating, 290-292
 - designing, 295-298
 - table creation, 292-298
 - installing, 14-16
- SQL Server Management Studio Express Edition, downloading, 586**
- "SQL Server Views," 292**
- SQL statements, 363**
 - DELETE, 365
 - INSERT, 364-365
 - UPDATE, 365-366
- SqlCommand class, 151**
- SqlDataSource Control's Markup listing (16.1), 362**
- SqlDataSource Web control, 307, 329-330**
 - adding to web pages, 307-313
 - choosing data connections, 308-310
 - configuring SELECT statement, 310-312
 - ConnectionString property, 313
 - declarative markup, 361-363
 - deleting data, 360-361
 - filtering data, 323-326
 - ID property, 313
 - inserting data, 360-361
 - markup, 313-314, 328-329
 - parameters, 362
 - SelectCommand property, 313
 - sorting data, 326-327
 - testing queries, 312-313, 327-328
 - updating data, 360-361
- SQLite, 285**
- Start Page, Visual Web Developer, 16-17**
- start tag (HTML), 26**
- starting debugger, 94**
- statements**
 - assignment statements, 104-105
 - Dim, 105, 109
 - If statements, 123-125
 - Else clause, 126-128
 - looping control statements, 128
 - Do...Loop loops, 130-131
 - For...Next loops, 128-130
 - modularizing statements, 131
 - code redundancy, 135-136
 - event handlers, 141
 - functions, 131
 - passing parameters, 136-138
 - returning values with, 138-141
 - subroutines, 131-135
- SELECT statement, 314**
 - configuring, 310-312
 - ORDER BY clause, 322-323
 - syntax, 314-315
 - viewing results of, 315-319
 - WHERE clause, 319-322
- SQL statements, 363**
 - DELETE, 365
 - INSERT, 364-365
 - UPDATE, 365-366
- static web pages, 9, 61**
 - dynamic web pages, compared, 22-23
 - serving, 9-10
- StaticEnableDefaultPopOutImage property (Menu control), 489**
- StaticItemFormatString property (Menu control), 490**
- StaticPopOutImageTextFormatString property (Menu Web control), 490**
- StaticPopOutImageUrl property (Menu control), 490**
- StaticSubMenuIndent property (Menu control), 490**
- Step Over command (Debug menu), 96**
- Stopping debugger, 96**

Strikeout subproperty (Label control Font property), 165

string data types, 108

strings, 288

- blank strings, GridView Web control, 378-379
- concatenation, 112-113
- connection strings, 309
- empty strings, 132
- variable value, inserting into, 113

structural definition, site maps, 471-472

Structured Query Language (SQL). See SQL (Structured Query)

style sheet files, 61

Subject property (MailDefinition), 539

subroutines, 152

- example, 132
- functions, compared, 142
- modularizing control structures, 131-135
- parameters, keywords, 137
- passing parameters to, 136-138
- reducing code redundancy, 135-136
- ShowWelcomeMessage, 135-136

subtraction (-) operator, 110

syntax

- functions, 139
- SQL SELECT statement, 314-315

T

table element (HTML), 33-37

tables (databases), 285-286

- adding data to, 298-301
- bit columns, displaying, 418-420
- columns, 285, 287-288
 - adding/removing, 302
 - auto-increment columns, 289
- Bit, 287
- datetime, 287
- float, 287
- Int, 287
- money, 287
- nvarchar(n), 287
- primary key columns, 288-290
- variable length character columns, 288

databases, creating, 292-298

deleting data

- DELETE statement, 365
- GridView Web control, 367-371
- SqlDataSource Web control, 360-361

inserting data

- DetailsView Web control, 385-387
- INSERT statement, 364-365
- SqlDataSource Web control, 360-361

modeling entities, 297-298

records, 285

- uniquely identifying, 295-297

retrieving data, 330

updating data

- GridView Web control, 373-384
- SqlDataSource Web control, 360-361
- UPDATE statement, 365-366

tasks, objects, 149

templated data Web controls, 441-442

- FormView, 456-458
 - custom templates, 460-462
- EditItemTemplate, 460
- EmptyDataTemplate, 460
- FooterTemplate, 460
- HeaderTemplate, 460
- InsertItemTemplate, 460
- ItemTemplate, 460
- PagerTemplate, 460
- paging records, 458-459
- templates, 459-460

ListView, 442-443

- adding templates, 445-450
- AlternatingItemTemplate, 446
- Configure ListView option, 443, 445-447
- EditItemTemplate, 446
- EmptyDataTemplate, 447
- GroupTemplate, 447, 463

templated data Web controls

- InsertItemTemplate, 447
- ItemSeparatorTemplate, 447
- ItemTemplate, 443-447
- LayoutTemplate, 447
- paging data, 450, 453-456
- sorting data, 450-452
- TemplateField, 381, 417, 437**
- TemplateField Has an ItemTemplate listing (16.2), 381**
- templates, 356, 379**
 - adding manually, 447-449
 - ASP.NET Web Site template, 547-549
 - FormView Web control, 459-462
 - EditItemTemplate, 460
 - EmptyDataTemplate, 460
 - FooterTemplate, 460
 - HeaderTemplate, 460
 - InsertItemTemplate, 460
 - ItemTemplate, 460
 - PagerTemplate, 460
 - ListView Web control
 - AlternatingItemTemplate, 446
 - EditItemTemplate, 446
 - EmptyDataTemplate, 447
 - GroupTemplate, 447, 463
 - InsertItemTemplate, 447
 - ItemSeparatorTemplate, 447
 - ItemTemplate, 447
 - LayoutTemplate, 447
 - paging, ListView Web control, 450, 453-456
 - sitewide page templates
 - creating, 504-505
 - designing, 502-506
 - footer region, 506
 - header region, 504-505
 - main region, 506
 - navigation region, 505-506
 - sorting, ListView Web control, 450-452
 - websites, Visual Web Developer, 58
- testing**
 - BMI (Body Mass Index) calculator, 190-192
 - financial calculator web page, 86-92
 - master pages, 515-516
 - queries, SqlDataSource, 327-328
 - RequiredFieldValidator Web control, 256-261
 - SQL queries, 312-313
 - web page, 21
- text**
 - displaying with Label Web control, 163-164
 - BackColor property, 166-167
 - border properties, 168-170
 - borders, 168
 - Font property, 163, 165, 169-171
 - ForeColor property, 166-167
 - formatting properties, 165-171
 - Height property, 170
 - setting properties programatically, 173
 - Text property, 164
 - Tooltip property, 170-171
 - Visible property, 170
 - Width property, 170
 - displaying with Literal Web control, 157-163
 - ID property, 159
 - Text property, 159-163
- text boxes**
 - collecting user input, 201-205
 - colors, 214-216
 - event handling, 203-205, 217
 - fonts, 214-216
 - limiting characters, 213-214
 - masked text boxes, 216
 - multiline text boxes, 217
 - creating, 205-208
 - password text boxes, creating, 205, 208-211
 - resizing, 212-213
- Text property (DropDownList), 229**
- Text property (Label), 164**
- Text property (Literal), 159**
 - setting programatically, 160-163
- Text property (RadioButton), 235-237**
- TextBox Web control, 201-205, 252**
 - BackColor property, 214-216
 - BorderColor property, 214
 - BorderStyle property, 214

- BorderWidth property, 214
 - colors, 214-215
 - Columns property, 212-214
 - event handling, 203-205
 - Font property, 214-216
 - FontColor property, 214
 - fonts, 214-215
 - ForeColor property, 214-216
 - MaxLength property, 214
 - multiline text boxes, 217
 - creating, 205-208
 - Password property, 208
 - password text boxes, 209
 - creating, 205, 208-211
 - resizing, 212-213
 - TextMode property, 206-207, 214
 - TextBox Web controls**
 - reading values in, 84-85
 - web pages, adding to, 77-79
 - TextLayout property (Login control), 543**
 - TextMode property, 206-207**
 - TextMode property (TextBox), 214**
 - Timer Web control, 569**
 - Toolbox (Visual Web Developer), 16, 36-37**
 - Tooltip property, Label Web control, 170-171**
 - TreeView Web control, 470**
 - CollapseImageTooltip property, 485
 - CollapseImageUrl property, 485
 - customizing appearance, 483-486
 - displaying site structure, 481-486, 488, 490
 - ExpandImageTooltip property, 485
 - ExpandImageUrl property, 485
 - HoverNodeStyle property, 484
 - ImageSet property, 485
 - LeafNodeStyle property, 484
 - LevelStyles property, 484
 - NodeIndent property, 485
 - nodes, 483
 - NodeStyle property, 484
 - NodeWrap property, 486
 - NoExpandImageUrl property, 485
 - ParentNodeStyle property, 484
 - RootNodeStyle property, 484
 - SelectedNodeStyle property, 484
 - ShowExpandCollapse property, 486
 - ShowLines property, 486
 - two-way data binding, 430**
 - types. See data types**
- U**
- UIs (user interfaces), 75**
 - adding Web controls, 81-82
 - buttons, adding, 79-80
 - choosing for web page, 75-76
 - creating, 76-82
 - TextBox Web controls, adding, 77-79
 - unary operator, 111
 - Underline subproperty (Label control Font property), 165**
 - Unicode characters, 288**
 - uniquely identifying records, 295-297**
 - UPDATE statement (SQL), 365-366**
 - UpdatePanel Web control, 558-559**
 - adding to Web pages, 559-562
 - multiple controls, 563-565
 - user action refresh, 564
 - UpdateProgress Web control, 558-559**
 - AssociatedUpdatePanelID, 567
 - DisplayAfter property, 567
 - displaying progress messages, 566-567
 - updating web.config, 576**
 - updating data**
 - SqlDataSource Web control, 360-361
 - UPDATE statement, 365-366
 - updating rules, GridView Web control, 373-384**
 - updating web.config settings, 588-591**
 - uploading website files, 576, 581-583**
 - user accounts, 521-528**
 - access rules, 529-532
 - allowing visitors to create, 533-540
 - anonymous users, 530

user accounts

- authentication, displaying
 - content based on, 545-547
 - classifying by role, 528-529
 - creating, 526-528
 - credentials, 522
 - customizing, 550
 - inactive user accounts,
 - creating, 540
 - logging in, 541-542
 - login page URL
 - specification, 544-545
 - logging out, 543-544
 - managing, 526-532
 - membership, configuring
 - websites to support, 522-526
 - sending email to new
 - accounts, 538-540
 - user action refresh, UpdatePanel Web control, 564**
 - user input**
 - collecting
 - check box, 238-242
 - classification, 222-223
 - drop-down lists, 224-232, 243
 - radio buttons, 233-238, 244
 - text boxes, 201-216
 - validation, 247-251
 - client-side validation, 258
 - CompareValidator, 261-267
 - comparison
 - validation, 249
 - CustomValidator, 274
 - data type validation, 249
 - formatting properties, 272-274
 - pattern validation, 250
 - range input
 - validation, 249
 - RangeValidator, 268-269
 - RegularExpressionValidator, 269-271, 277
 - required field input
 - validation, 249
 - RequiredFieldValidator, 253-261
 - sample web page, 251-253
 - server-side validation, 258
 - Validation Summary Web control, 275
 - Web Forms
 - collecting, 186-192
 - processing, 192-194
 - web pages, collecting, 180-187
 - user input, collecting**
 - drop-down lists
 - properties, 232
 - rearranging order of items, 230
 - overview, 180
 - text boxes
 - colors, 214-215
 - fonts, 214-215
 - password text boxes, 209
 - user interfaces, financial calculator web page**
 - Label Web control, 82
 - source code, 85-86
 - testing, 86-92
 - TextBox Web controls, 84-85
 - viewing source code, 87-91
 - user interfaces (UIs). See UIs (user interfaces)**
 - User's Ice Cream Preferences Are Displayed listing (11.2), 241**
 - UserName property (CreateUserWizard), 536**
 - UserNameLabelText property (CreateUserWizard), 536**
 - UserNameRequiredErrorMessage property (CreateUserWizard), 536**
 - Using the CustomValidator Control (article), 275**
 - "Using the UpdatePanel," 564**
 - "Utilizing Client-Side Script to Confirm Deletions," 388**
- ## V
- validation (user input), 247-251**
 - client-side validation, 258
 - CompareValidator, 261-265
 - comparing input, 265-267
 - ErrorMessage
 - property, 263
 - GreaterThanEqual
 - property, 262
 - Operator property, 262
 - ValueToCompare
 - property, 263
 - comparison validation, 249
 - CustomValidator, 274

- data type validation, 249
- formatting properties, 272-274
- pattern validation, 250
- range input validation, 249
- RangeValidator, 268-269
- RegularExpressionValidator, 269-271, 277
- required field input validation, 249
- RequiredFieldValidator, 253-254
 - ControlToValidate property, 253-255
 - features, 261
 - specifying error messages, 255-256
 - specifying validation, 253-255
 - testing, 256-261
- sample web page, 251-253
- server-side validation, 258
- Validation Summary Web control, 275
- Validation Summary Web control, 275**
- validation Web controls, 250-251**
 - CompareValidator, 247, 250, 261-262, 264-265
 - comparing input, 265-267
 - ErrorMessage property, 263
 - GreaterThanEqual property, 262
 - Operator property, 262
 - ValueToCompare property, 263
- CustomValidator, 274
- formatting properties, 272-274
- RangeValidator, 250, 268-269
- RegularExpressionValidator, 250, 269-271, 277
- RequiredFieldValidator, 250, 253-254
 - ControlToValidate property, 253-255
 - features, 261
 - specifying error messages, 255-256
 - specifying validation, 253-255
 - testing, 256-261
- sample web page, 251-253
- Validation Summary, 275
- Value property (DropDownList), 229**
- values**
 - returning, functions, 138-141
 - variables
 - assigning values to, 104-105
 - inserting into strings, 113
- ValueToCompare property (CompareValidator), 263**
- varchar data type, 288**
- variable length character columns, 288**
- variables, 117-118**
 - data types, casting, 116-117
 - declaring, assignment performance, 110
 - immutable values, 104
- mutable values, 104
- strings, inserting values, 113
- Visual Basic
 - assigning values to, 104-105
 - data types, 104-109
 - declaring, 103-105
 - naming, 103-106
- viewing**
 - source code, financial calculator web page, 87-91
 - SQL SELECT statement results, 315-319
 - Visual Web Developer windows, 68-69
- views, Visual Web Developer, 29-31**
- Visible property (Label), 170**
- VisibleWhenLoggedIn property (Login), 543**
- visitors, user accounts, allowing to create, 533-540**
- Visual Basic, 53, 101-102**
 - apostrophe ('), 92
 - classes
 - constructors, 150-151
 - creating, 155
 - definition, 148
 - MailMessage, 155
 - overloaded, 151
 - SmtpClient, 155
 - SqlCommand, 151
 - conditional control structures, 123-125
 - Else clause, 126-128

Visual Basic

- control structures,
 - 121-122, 142
 - loops, 131
 - subroutines, 132, 136
 - CType function, 117
 - data types, rules, 116-117
 - Dim statement, 109
 - event handlers, 141
 - creating, 153-154
 - naming, 141
 - functions
 - calling, 143
 - syntax, 139
 - versus subroutines, 142
 - looping control
 - structures, 128
 - Do...Loop loops, 130-131
 - For...Next loops, 128-130
 - methods, 150
 - calling, 149
 - modularizing control
 - structures, 131
 - code redundancy, 135-136
 - event handlers, 141
 - functions, 131
 - passing parameters, 136-138
 - returning values, 138-141
 - subroutines, 131-135
 - objects, 147, 154-155
 - calling methods, 152-153
 - classes, 148-149
 - common tasks, 149
 - creating, 150-151
 - creating event handlers, 149
 - definition, 148
 - instances, 148
 - instantiation, 149
 - role of, 149
 - setting properties, 151-152
 - setting property values, 149
 - operators, 110
 - arithmetic operators, 110-111
 - assignment operators, 113-115
 - comparison operators, 111-112, 320
 - concatenation operator, 112-113
 - logical operators, 320
 - precedence, 111
 - subroutines versus functions, 142
 - variables, 117-118
 - assigning values to, 104-105
 - data types, 104-109
 - declaring, 103-105
 - naming, 103-106
- Visual Web Developer, 8, 57, 70.**
- See also Web controls**
- AutoRecover feature, 66
 - colors, 67
 - community, 69
 - CSS Properties window, 16
 - customizing, 65-68
 - Default.aspx file, 19
 - Design view, 29-31
 - editing HTML content, 33-40
 - Empty Web Site template, 17
 - exercises, 54-55, 72
 - fonts, 67
 - Help, 69-70
 - installing, 14-16
 - Manage Styles window, 16-17
 - repositioning windows, 20
 - settings, saving, 68
 - site maps, creating, 474
 - Solution Explorer, 16
 - adding content to websites, 62-64
 - Source view, 29, 31
 - Split view, 30-31, 33
 - Start Page, 16-17
 - Toolbox, 16, 37
 - viewing SQL query results, 315-323
 - web pages, creating HTML portion, 29-43
 - websites
 - adding content to, 62-64
 - creating, 17-21, 58-60
 - deleting content, 64-65
 - locations, 58-60
 - moving content, 64-65
 - opening existing, 60-61
 - renaming content, 64-65
 - source code programming language, 60
 - templates, 58
 - testing, 21-22

- windows
 - closing, 69
 - moving, 68-69
 - resizing, 68
 - viewing, 68-69
 - WYSIWYG Design view, 76
 - Visual Web Developer**
 - Toolbox, 306
- ## W-Z
- “Walkthrough: Debugging Web Pages in Visual Web Developer,” 96
 - Watch window (debugger), 96
 - Web applications, building, 577-581
 - web browsers, Ajax compatibility, 568
 - Web controls, 29, 157, 191
 - BooksDataSource, 442
 - ChangePassword, 550
 - CheckBox, 238-242
 - choosing, 223
 - ContentPlaceHolder, 498-499
 - adding to master pages, 500-506
 - default master pages, 512
 - CreateUserWizard, 533-535
 - customizing, 535-537
 - data binding
 - declarative markup, 435-436
 - specifying for, 430-434
 - data source Web controls, 305-306
 - AccessDataSource, 307
 - SqlDataSource, 307-314, 323-330
 - data Web controls, 333-337
 - DetailsView control, 347-351
 - GridView control, 337-357
 - DetailsView
 - customizing Insertable
 - DetailsView, 385-387
 - displaying bit columns, 418-420
 - displaying hyperlinks, 420-425, 436
 - displaying images, 425-426
 - fields, 416-425, 436
 - inserting data, 385-386
 - DropDownList, 224
 - adding items to, 224-225
 - adding to web pages, 225-232
 - BackColor property, 232
 - copying, 243
 - Font property, 232
 - ForeColor property, 232
 - populating from
 - databases, 243
 - rearranging order of items, 230
 - Selected property, 228
 - SelectedItem property, 231
 - Text property, 229
 - Value property, 229
 - GridView, 366
 - allowing users to delete data, 367-371
 - blank strings, 378-379
 - changing data source, 346
 - creating editable
 - GridViews, 371-373
 - displaying confirmation messages, 388
 - customizing editing interface, 373-384
 - displaying bit columns, 418-420
 - displaying hyperlinks, 420-425, 436
 - displaying images, 425-426
 - Edit field, 373
 - editing fields, 377-378
 - fields, 416-425, 436
 - marking fields as
 - Read-Only, 375-377
 - nulls, 378-379
 - updating rules, 373-384
 - Label, 163, 172-173
 - borders, 168
 - displaying text, 163-173
 - financial calculator web page, 82
 - list Web controls, 394-395
 - binding data to, 395-397
 - CheckBoxList, 406-411
 - DropDownList, 401-406, 412
 - dynamically populating, 397-398

Web controls

- programmatically responding to changed selections, 398-401
- RadioButtonList, 406-412
- Literal Web control, 172-173
 - displaying text, 157-163
- Login, 541-545
 - customizing, 542
- Menu, displaying site structure, 487-491
- navigation Web controls
 - Menu, 470
 - SiteMapPath, 470
 - TreeView, 470
- PasswordRecovery, 550
- programmatically working with, 49-52
- properties, setting, 52
- RadioButton, 233-238, 244
 - GroupName property, 235-237
 - Text property, 235-237
- ScriptManager, 558-560
- SiteMapPath
 - customizing appearance, 478-481
 - displaying breadcrumbs, 476-478
 - nodes, 479
 - path separators, 479
- SqlDataSource
 - declarative markup, 361-363
 - deleting data, 360-361
 - inserting data, 360-361
 - parameters, 362
 - updating data, 360-361
- templated data Web controls, 441-442
 - FormView, 456-462
 - ListView, 442-456
- TextBox, 201-205, 252
 - BackColor property, 214-216
 - BorderColor property, 214
 - BorderStyle property, 214
 - BorderWidth property, 214
 - colors, 214-215
 - Columns property, 212-214
 - event handling, 203-205
 - Font property, 214-216
 - FontColor property, 214
 - fonts, 214-215
 - ForeColor property, 214-216
 - MaxLength property, 214
 - multiline text boxes, 205-208, 217
 - Password property, 208
 - password text boxes, 205, 208-211
 - reading values in, 84-85
 - resizing, 212-213
 - TextMode property, 206-207, 214
- Timer, 569
- TreeView
 - customizing appearance, 483-486
 - displaying site structure, 486-490
- UpdatePanel, 558-559
 - adding to Web pages, 559-562
 - multiple controls, 563-565
 - user action refresh, 564
- UpdateProgress, 558-559
 - displaying progress messages, 566-567
- validation Web controls, 250-251
 - CompareValidator, 250, 261-267
 - CustomValidator, 274
 - formatting properties, 272-274
 - RangeValidator, 250, 268-269
 - RegularExpressionValidator, 250, 269-271, 277
 - RequiredFieldValidator, 250, 253-261
 - sample web page, 251-253
 - Validation Summary, 275
- web pages, adding to, 37-39, 81-82
 - when to use, 53
- Web Forms, 77, 179, 186-187, 197-198**
 - BMI Calculator, 188-190
 - testing, 190-192
 - writing, 195-197
 - postback forms, 185-187, 192, 198
 - refreshing web pages, 198

- redirect forms, 185-187
- user input, processing, 192-194
- web pages. See websites**
- web requests, 9-10**
- web servers, 9**
 - dynamic web pages, serving, 10-12
 - hosting web pages, 59
 - Development Web Server, 12-13
 - IIS (Internet Information Server), 22-23
 - static web pages, serving, 9-10
 - web requests, 10
- web-hosting companies**
 - choosing, 70, 574-576
 - minimum requirements, 575
 - web sites, deploying to, 13
- web.config file, updating, 576, 588-591**
- Web.sitemap file, 473**
- Website Administration Tool, 522-523, 592**
- websites. See also user accounts**
 - adding content, Visual Web Developer, 62-64
 - buttons, adding to, 79-80
 - check boxes, 238-242
 - determining checks, 239-242
 - collecting user input, 180
 - form element, 182-187
 - input element, 180-182
 - Web Forms, 186-192
 - content pages, 496
 - creating, 507-510
 - creating, 17-20
 - Visual Web Developer, 58-60
 - data Web controls, 333-337
 - DetailsView control, 347-351
 - GridView control, 337-347, 351-357
 - deleting content, Visual Web Developer, 64-65
 - deployment, 573, 592-593
 - database replication, 583-588
 - process, 576-590
 - remote websites, 590-591
 - updating web.config settings, 588-590
 - uploading website files, 581-583
 - web-hosting companies, 574-576
 - design requirements, 73-74
 - features, 74-75
 - UIs (user interfaces), 75-76
 - domain names, 574
 - drop-down lists, 224
 - adding items to, 224-225
 - adding to, 225-232
 - copying, 243
 - populating from databases, 243
 - DropDownList.aspx, 230-231
 - dynamic web pages, 7-9, 22, 61
 - serving, 10-12
 - files
 - adding, 64
 - types, 61-62
 - uploading, 576
 - financial calculator web page
 - Label Web control, 82
 - source code, 85-86
 - testing, 86-92
 - TextBox Web controls, 84-85
 - viewing source code, 87-91
 - writing, 82-83
 - hosting, 12-13, 59
 - web-hosting companies, 70
 - HTML, 25-29
 - portion, creating, 29-43
 - image files, 61
 - LiteralTime.aspx, 161-162
 - locations, Visual Web Developer, 58-60
 - logging in, 541-542
 - login page URL specification, 544-545
 - logging out, 543-544
 - master pages, 63, 492, 495-499, 544
 - creating, 500-506
 - default content in, 510-512
 - inheriting, 509-510

websites

- source code, 513-516
- testing, 515-516
- moving, 71
- moving content, Visual Web Developer, 64-65
- opening, Visual Web Developer, 60-61
- PrettyTextBox.aspx, 215
- processing user input, Web Forms, 192-194
- programming language, choosing, 60
- radio buttons, 233-238, 244
- refreshing postback forms, 198
- renaming content, Visual Web Developer, 64-65
- sending email from, 155
- sign-in pages, 541
- site maps, 469
 - adding, 472-474
 - building, 474-476
 - customizing appearance, 478-481
 - displaying breadcrumbs, 476-478
 - displaying site structure, 481-491
 - providers, 492
 - structural definition, 471-472
- site navigation, 469-471
- SMTP settings, configuring, 532-533
- source code
 - debugging, 93-96
 - examining, 44-52, 92-93
 - server-side code, 45
- SqlDataSource Web control, 329-330
 - adding to, 307-313, 323-329
 - static, 9, 22, 61
 - serving, 9-10
 - tables, adding, 33-37
 - templates, Visual Web Developer, 58
 - testing, 21
 - text boxes
 - colors, 214-216
 - event handling, 203-205, 217
 - fonts, 214-216
 - limiting characters, 213-214
 - masked text boxes, 216
 - multiline text boxes, 217
 - resizing, 212-213
- Uls (user interfaces)
 - creating, 76-82
 - TextBox, 77-79
- UpdatePanel, adding to, 559-562
- user accounts, 521-528
 - credentials, 522
- user input
 - collecting, 180, 201-216, 222-244
 - validating, 247-277
- web controls, adding to, 37-39, 81-82
- Web forms, 77
- web-hosting companies, deploying to, 13
- writing source code for, 84-86
- WHERE clause**
 - queries, testing, 327-328
 - SELECT statement, 319-322
- WHERE filter expression, wildcards, 426-429, 437**
- whitespace, HTML, 27-28**
- width, text boxes, specifying, 212-213**
- Width property (Label), 170**
- wildcards, WHERE filter expressions, 426-429, 437**
- windows, Visual Web Developer**
 - closing, 69
 - moving, 68-69
 - resizing, 68
 - viewing, 68-69
- Windows authentication, 523**
- wizards**
 - Configure Data Source Wizard, 308
 - CreateUserWizard web control, 533-537
 - Database Publishing Wizard, 584-588
- writing**
 - BMICalculator.aspx, 195-197
 - Source code for websites, 84-86
- "Writing a Stored Procedure," 292**
- WYSIWYG Design view, 76**
- XML (eXtensible Markup Language), 470**
 - formatting rules, 475
- XML files, 307, 470**