1.18   Tour of the Book

In this section, we take a tour of the subjects you will study in *Internet and World Wide Web How to Program, Second Edition*. Many of the chapters end with an Internet and World Wide Web Resources section that provides a listing of resources through which you can enhance your knowledge and use of the Internet and the World Wide Web. In addition, you may want to visit our Web site **www.deitel.com** for additional resources.

Chapter 1—Introduction to Computers and the Internet
In Chapter 1, we present historical information about computers and computer programming and introductory information on the Internet and the World Wide Web. We also overview the technologies and concepts discussed in the remaining chapters of the book.

Chapter 2—Microsoft® Internet Explorer 5.5[2]
Prior to the explosion of interest in the Internet and the World Wide Web, if you heard the term *browser,* you probably thought about browsing at a bookstore. Today "browser" has a whole new meaning—an important piece of software that enables you to view Web pages. The two most popular browsers are Microsoft's Internet Explorer and Netscape's Communicator. Throughout this book, we use Internet Explorer 5.5, but we provide a solid introduction to Netscape Communicator 6 at **www.deitel.com**. Using tools included with Internet Explorer, we demonstrate how to use the Web. These tools include, but are not limited to, the Web browser, e-mail, newsgroups (i.e., where users can post messages on a variety of topics to the general public) and instant messaging, which allows users to communicate over the Internet in real time. This chapter shows readers unfamiliar with the World Wide Web how to browse the Web with Internet Explorer. We demonstrate several commonly used features for searching the Web, keeping track of the sites you visit and transferring files between computers. We also discuss several programs included with Internet Explorer. We demonstrate sending and receiving e-mail, and using Internet newsgroups with Microsoft *Outlook Express*. We demonstrate MSN *Instant Messenger*, which enables almost instant conferencing with friends, family and coworkers. We demonstrate Microsoft *NetMeeting* and Microsoft *Chat* for having live meetings and discussions with other people on the Internet. The chapter concludes with a discussion of browser *plug-ins* that provide access to the ever-increasing number of programs and features that make browsing more enjoyable and interactive.

Chapter 3—Adobe® PhotoShop® Elements
The Internet and World Wide Web are rich in multimedia content. Web pages contain colorful graphics, sounds and text. Graphics are an essential element of Web-page design that convey visual information about a site's contents. In this chapter, we introduce *Adobe PhotoShop Elements*, a graphics software package that contains an extensive set of tools and features for creating high-quality graphics and animations. These tools and features include filters for applying special effects and screen capturing for taking "snap shots" of the screen. Chapter examples demonstrate creating title images for a Web page, creating a navigation bar that contains a series of buttons used to connect a Web site's pages and manipulating images by using advanced photographic effects. We focus on creating and

---

2.   We provide a comparable chapter on Netscape Communicator 6 at **www.deitel.com**.

manipulating the two most popular image formats used in Web documents: *Graphics Interchange Format (GIF)* and *Joint Photographic Expert Group (JPEG)* files. [*Note*: Readers can download a 30-day evaluation copy of PhotoShop Elements from **www.adobe.com/support/downloads**. The chapter examples were developed using that version of PhotoShop Elements.]

Chapter 4—Introduction to XHTML: Part 1

In this chapter, we unlock the power of Web-based application development by introducing *XHTML*—the *Extensible Hypertext Markup Language*. XHTML is a *markup language* for identifying the elements of an XHTML document (or Web page) so that a browser can render (i.e., display) that document on a computer screen. We introduce basic XHTML Web-page creation using a technique we call the live-code™ approach. Every concept is presented in the context of a complete working XHTML document. We render each working example in Internet Explorer and show the screen outputs. We present many short Web pages that demonstrate XHTML features. Later chapters introduce more sophisticated XHTML techniques, such as *tables*, which are useful for formatting information retrieved from a database. We introduce XHTML *tags* and *attributes*, which describe the document's information. A key issue when using XHTML is the separation of the *presentation of a document* (i.e., how the document is rendered on the screen by a browser) from the *structure of the information in that document* (i.e., the information the document contains). This chapter introduces our in-depth discussion of this issue. As the book proceeds, you will be able to create appealing and powerful Web pages and Web-based applications. Other topics in this chapter include incorporating text, images and special characters (such as copyright and trademark symbols) into an XHTML document, validating an XHTML document to ensure that it is written correctly, placing information inside *lists*, separating parts of an XHTML document with horizontal lines (called *horizontal rules*) and linking to other XHTML documents on the Web. In one of the chapter exercises, we ask readers to mark up their resume with XHTML.

Chapter 5—Introduction to XHTML: Part 2

In this chapter, we discuss more substantial XHTML elements and features. We demonstrate how to present information in *tables* and how to gather user input. We explain and demonstrate *internal linking* and *image maps* to make Web pages more navigable and how to use *frames* to display multiple XHTML documents in a browser. *XHTML forms* are one of the most important features introduced in this chapter—forms display information to the user and accept user input. By the end of this chapter, readers should be familiar with the most popular XHTML tags and features used to create Web sites.

Chapter 6—Cascading Style Sheets (CSS)

Web browsers control the appearance (i.e., the rendering) of every Web page. For instance, one browser may render an **h1** (i.e., a large heading) element in an XHTML document differently than another browser. With the advent of *Cascading Style Sheets (CSS)*, Web developers can control the appearance of their Web pages. CSS allows Web developers to specify the style of their Web page's elements (spacing, margins etc.) separately from the structure of their pages (section headers, body text, links etc.). This *separation of structure from content* allows greater manageability and makes changing document styles easier and

faster. We introduce *inline*, *embedded* and *external* style sheets. Inline style sheets are applied to individual XHTML elements, embedded style sheets are entire style sheets placed directly inside an XHTML document and external style sheets are style sheets located outside an XHTML document.

Chapter 7—JavaScript:[3] Introduction to Scripting

Chapter 7 presents our first JavaScript *programs*[4] (also called *scripts*). Scripting helps Web pages "come alive." Web developers dynamically manipulate Web-page elements through scripting as clients browse Web pages. Chapters 7–12 present JavaScript, which is then used in Chapters 13–18 to manipulate Web-page content. We present the key fundamental computer-science concepts of JavaScript at the same depth as we do in our other books on conventional programming languages (such as C, C++, C#, Java and Visual Basic), but in the exciting context of the Internet and World Wide Web. Using our live-code™ approach, we present every concept in the context of a working JavaScript program that is immediately followed by the screen output. The chapter introduces nonprogrammers to basic programming concepts and constructs. The scripts in this chapter illustrate how to output text to a browser and how to obtain user input through the browser. Some of the input and output is performed using the browser's capability to display predefined *graphical user interface (GUI)* windows (called *dialogs*). This allows nonprogrammers to concentrate on fundamental programming concepts and constructs rather than on GUI components and on GUI *event handling*. Chapter 7 also provides detailed treatments of *decision making* and *arithmetic operations*.

Chapter 8—JavaScript: Control Structures 1

Chapter 8 focuses on the program-development process. The chapter discusses how to develop a working JavaScript program from a *problem statement* (i.e., a *requirements document*). We show the intermediate steps using, a program development tool called *pseudocode*. The chapter introduces some simple control structures used for decision making (**if** and **if**/**else**) and repetition (**while**). We examine countercontrolled repetition and sentinel-controlled repetition and introduce the increment, decrement and assignment operators. Simple flowcharts illustrate graphically the flow of control through each of the control structures. This chapter helps the student develop good programming habits in preparation for the more substantial programming tasks in the remainder of the book.

Chapter 9—JavaScript: Control Structures 2

Chapter 9 discusses much of the material JavaScript has in common with the C programming language, especially the *sequence*, *selection* and *repetition* control structures. Here, we introduce one additional control structure for decision making (**switch**) and two ad-

---

3. Netscape created JavaScript; the Microsoft version is called JScript. The two scripting languages are similar. Netscape, Microsoft and other companies cooperated with the European Computer Manufacturer's Association (ECMA) to produce a universal, client-side scripting language, which is referred to as ECMA-262. JavaScript and JScript each conform to this standard.

4. The book's JavaScript examples execute in Microsoft Internet Explorer 5.5. We have tested these examples on the following clients: Internet Explorer 5.5, Internet Explorer 6 Beta and Netscape Communicator 6. For those few examples that do not execute in Netscape Communicator 6, we have (when possible) created Netscape Communicator 6 equivalent examples. These examples and the test results are available at **www.deitel.com**.

ditional control structures for repetition (**for** and **do/while**). This chapter also introduces several operators that allow programmers to define complex conditions in their decision-making and repetition structures. The chapter uses flowcharts to illustrate the flow of control through each of the control structures, and concludes with a summary that enumerates each of the structures. The techniques discussed in this chapter and in Chapter 10 constitute a large part of what has been traditionally taught in universities under the topic of structured programming.

Chapter 10—JavaScript: Functions

Chapter 10 takes a deeper look inside scripts. Scripts contain data called *global* (or *script-level*) *variables* and executable units called *functions*. We discuss JavaScript functions, programmer-defined functions and *recursive* functions (i.e., functions that call themselves). The techniques presented in Chapter 10 are essential to produce properly structured programs, especially large programs that Web developers are likely to build in real-world, Web-based applications. The *divide-and-conquer* strategy is presented as an effective means for solving complex problems by dividing them into simpler, interacting components. The chapter offers a solid introduction to recursion and includes a table summarizing the many recursion examples and exercises in Chapters 10–12. We introduce *events* and *event handling*—elements required for programming graphical user interfaces (GUIs) in XHTML forms. Events are notifications of state changes, such as button clicks, mouse clicks, pressing keyboard keys, etc. JavaScript allows programmers to respond to various events by coding functions called *event handlers*. This begins our discussion of *event-driven programming*—the user drives the program by interacting with GUI components (causing *events such as mouse clicks*), and the scripts respond to the events by performing appropriate tasks (*event handling*). The event-driven programming techniques introduced here are used in scripts throughout the book. Dynamic HTML event handling is introduced in Chapter 14. Chapter 10 contains a rich set of exercises that include the Towers of Hanoi, computer-aided instruction and a guess-the-number game.

Chapter 11—JavaScript: Arrays

Chapter 11 explores the processing of data in lists and tables of values. We discuss the structuring of data into *arrays*, or groups, of related data items. The chapter presents numerous examples of both single-subscripted arrays and double-subscripted arrays. It is widely recognized that structuring data properly is as important as using control structures effectively in the development of properly structured programs. Examples in the chapter investigate various common array manipulations, searching arrays, sorting data and passing arrays to functions. This chapter introduces JavaScript's **for**/**in** control structure, which interacts with collections of data stored in arrays. The end-of-chapter exercises include a variety of interesting and challenging problems, such as the *Sieve of Eratosthenes* and the design of an airline reservations system. The chapter exercises also include a delightful simulation of the classic race between the tortoise and the hare.

Chapter 12—JavaScript: Objects

This chapter discusses *object-based programming* with JavaScript's built-in objects. The chapter introduces the terminology of objects. We overview the methods (functions associated with particular objects) of the JavaScript **Math** object and provide several examples of JavaScript's string-, date- and time-processing capabilities with the **String** and **Date**

objects. An interesting feature of the **String** object is a set of methods that help a programmer output XHTML from a script by enclosing strings in XHTML elements. The chapter also discusses JavaScript's **Number** and **Boolean** objects. Many of the features discussed in this chapter are used in Chapters 13–18 to illustrate that every XHTML element is an object that can be manipulated by JavaScript statements. Many challenging, yet entertaining, string-manipulation exercises are included.

Chapter 13—Dynamic HTML:[5] DHTML Object Model and Collections
A massive switch is occurring in the computer industry. The procedural programming style used since the inception of the industry is being replaced by the object-oriented style of programming. The vast majority of new software efforts use object technology in one form or another. The scripting languages we discuss in this book usually manipulate existing objects by sending messages that either inquire about the objects' attributes or ask the objects to perform certain actions. In this chapter, we continue the discussion of object technology by presenting Microsoft's Dynamic HTML object model. As Internet Explorer downloads a Web page from a server, it converts each element to an object. Objects store data (their attributes) and perform functions (their methods). Through scripting languages such as JavaScript, you can write commands that *get* or *set* (i.e., read or write) an object's attributes. You can also write commands that invoke an object's methods. The chapter exercises provide the opportunity to program the classic "15-puzzle" game.

Chapter 14—Dynamic HTML: Event Model
We have discussed how scripting can control XHTML pages. Dynamic HTML includes *event models* that enable scripts to respond to user actions. This allows Web applications to be more responsive and user friendly, and can it reduce server load—a performance concern we discuss in Chapters 21–31 on server-side programming. With the event model, scripts can respond to a user moving or clicking the mouse, scrolling up or down the screen or entering keystrokes. Content becomes more dynamic, while interfaces become more intuitive. We discuss how to use the event model to respond to user actions. We provide examples of event handling, which range from mouse capture to error handling to form processing. For example, we call the **onreset** event to confirm that a user wants to reset the form (i.e., the GUI in which the user inputs data). For one of the chapter exercises, the reader creates an interactive script that displays an image alongside the mouse pointer. When the mouse pointer is moved, the image moves with it.

Chapter 15—Dynamic HTML: Filters and Transitions
Internet Explorer includes a set of filters that allow developers to perform complex image transformations entirely in the Web browser without the need for additional downloads from a Web server. Filters are scriptable, so the developer can create stunning, customized animations with a few lines of client-side JavaScript. We introduce the **fliph** and **flipv** filters, which mirror text and images horizontally and vertically. We explain the **gray**,

---

5. Microsoft Dynamic HTML and Netscape Dynamic HTML are incompatible. In this book, we focus on Microsoft Dynamic HTML. We have tested all of the Dynamic HTML examples in Internet Explorer 5.5 and Netscape Communicator 6. All of these examples execute in Internet Explorer, but do not execute in Netscape Communicator 6. We have posted the testing results at **www.deitel.com**. In this book, we also present Macromedia® Flash™, which executes in Internet Explorer and Netscape Communicator 6.

xray and **invert** filters, which all apply simple transformations to images. We introduce many of the filters that apply effects such as shadows, transparency gradients and distortions. Internet Explorer enables *transitions* that are similar to transitions between slides in PowerPoint-like presentations. The **revealTrans** filter applies visual effects such as *box in*, *circle out*, *wipe left*, *vertical blinds*, *checkerboard across*, *random dissolve*, *split horizontal in*, *strips right up* and *random bars horizontal*. This chapter also introduces the **blendTrans** filter, which allows you to fade in or fade out of an XHTML element over a set interval.

Chapter 16—Dynamic HTML: Data Binding with Tabular Data Control
This is one of the most important chapters in the book for people who want to build substantial, real-world Web-based applications. Businesses thrive on data, and Dynamic HTML helps Web developers build data-intensive applications. With *data binding*, data does not reside exclusively on the server. Data are sent from the server to the client, and all subsequent manipulations of the data occur on the client. Data can be maintained on the client in a manner that distinguishes the data from the XHTML markup. Manipulating data on the client improves performance by eliminating server activity and network delays. Once data is available on the client, the data can be *sorted* (i.e., arranged into ascending or descending order) and *filtered* (i.e., selected according to some criterion) in various ways. We present examples of each of these operations. To bind external data to XHTML elements, Internet Explorer employs software capable of connecting the browser to live data sources, known as *Data Source Objects (DSOs)*. Several DSOs are available in Internet Explorer—in this chapter, we discuss the most popular DSO, *Tabular Data Control (TDC)*.

Chapter 17—Dynamic HTML: Structured Graphics ActiveX Control
Although high-quality content is important to a Web site, it does not attract or maintain visitors' attention like eye-catching, animated graphics. This chapter explores the *Structured Graphics ActiveX Control* included with Internet Explorer. The Structured Graphics Control is a Web interface for the *DirectAnimation* subset of Microsoft's *DirectX* software. DirectAnimation is used in many popular video games and graphical applications. This control allows you to create complex graphics containing lines, shapes, textures and fills. In addition, scripting allows the graphics to be manipulated dynamically. The exercises at the end of the chapter ask the reader to create three-dimensional shapes and rotate them.

Chapter 18—Dynamic HTML: Path, Sequencer and Sprite ActiveX Controls
In this chapter, we discuss three additional DirectAnimation ActiveX controls available for Internet Explorer: the *Path Control*, the *Sequencer Control* and the *Sprite Control*. Each of these controls allow Web developers to add animated multimedia effects to Web pages. The Path Control allows the user to determine the positioning of elements on the screen. This is more elaborate than CSS absolute positioning, because the user can define lines, ovals and other shapes as paths along which objects move. Every aspect of motion is controllable through scripting. The Sequencer Control performs tasks at specified time intervals. This is useful for presentationlike effects, especially when used with the transitions discussed in Chapter 15. The Sprite Control creates Web animations. We also discuss, for comparison purposes, animated GIFs—another technique for producing Web-based animations.

Chapter 19—Macromedia® Flash™: Creating Interactive Web Pages

Macromedia Flash[6] is a cutting-edge multimedia application that creates interactive content for the World Wide Web. Through hands-on examples, this chapter shows how to add interactivity, sound and animation to Web sites, while teaching the fundamentals of Macromedia Flash and *ActionScript*, Flash's scripting language. The chapter examples include creating interactive buttons, animated banners and animated splash screens (called *animation preloaders*). The exercises ask the reader to create a navigation bar, a spotlight effect and a morphing effect. The morphing effect exercise in particular is a wonderful illustration of the power of Flash. Readers will enjoy watching text transform into a shape and back.

Chapter 20—Extensible Markup Language (XML)

Throughout the book, we have been emphasizing XHTML. This language derives from SGML (Standardized General Markup Language), which became an industry standard in 1986. SGML is employed in publishing applications worldwide, but it has not been incorporated into mainstream computing and information technology curricula. Its sheer size and complexity limit its use beyond heavy-duty, industrial-strength applications. The Extensible Markup Language (XML) is an effort to make SGML-like technology available to a much broader community. XML, a condensed subset of SGML, contains additional features for usability. XML differs in concept from XHTML. XHTML is a markup language, and XML is a language for *creating* markup languages. XML enables document authors to create their own markup for virtually any type of information. As a result, document authors use this extensibility to create entirely new markup languages to describe specific types of data, including mathematical formulas, chemical molecular structures, music and recipes. Markup languages created with XML include XHTML (Chapters 4 and 5), MathML (for mathematics), VoiceXML™ (for speech), SMIL™ (the Synchronized Multimedia Integration Language, for multimedia presentations), CML (Chemical Markup Language, for chemistry) and XBRL (Extensible Business Reporting Language, for financial data exchange). XML is a technology created by the World Wide Web Consortium for describing data in a portable format. XML is one of most important technologies in industry today and is being integrated into almost every field. Every-day, companies and individuals are finding new and exciting uses for XML. In this chapter, we present examples that illustrate the basics of marking up data using XML. We demonstrate several XML-derived markup languages, such as MathML, CML, *XML Schema* (for checking an XML document's grammar), *XSLT (Extensible Stylesheet Language Transformations*, for transforming an XML document's data into an XHTML document) and Microsoft's *BizTalk™* (for marking up business transactions). The reader interested in a deeper treatment of XML may want to consider our book, *XML How to Program*.

Chapter 21—Web Servers (IIS, PWS and Apache)

Through Chapter 20, we have focused on the client side of Web-based applications. Chapters 21–31 focus on the server side, discussing many technologies crucial to implementing

---

6. Many browsers, including Internet Explorer 5.5 and Netscape Communicator 6, support Macromedia Flash content. All of the functionality in this chapter has been tested on, and properly works on both Internet Explorer 5.5 and Netscape Communicator 6. Some Web developers prefer Flash to Dynamic HTML. Some use both.

successful Web-based systems. A Web server is part of a *multitiered application*—sometimes referred to as an *n*-tier application. A three-tier application contains a *data tier (bottom tier)*, *middle tier* and *client tier (top tier)*. The bottom tier is an organization's database. The middle tier receives client requests from the top tier, references the data stored in the bottom tier and sends the requested information to the client. The client tier renders a Web page and executes any scripting commands contained in the Web page. A crucial decision in building Web-based systems is which Web server to use. The *Apache Web Server* and Microsoft *Internet Information Services (IIS)* are the two most popular Web servers used in industry. Each of these is an "industrial-strength" server designed to handle the high volumes of transactions that occur in real-world systems. They require considerable system resources and administrative support. To help people enter the world of server programming, Microsoft provides *Personal Web Server (PWS)*—a scaled-down version of IIS. In this chapter, we provide a brief introduction to IIS, PWS and Apache. We provide installation instructions for these Web servers at **www.deitel.com**. We discuss how to request XHTML, ASP, Perl, Python and PHP documents from these Web servers when using Internet Explorer. The chapter concludes by listing some additional Web servers that are available on the Internet. *[Note: The world of server software is complex and evolving quickly. Our goal in this chapter is to give you a "handle" on setting up and using server-side software. Deitel & Associates, Inc., does not provide software support for these servers. We suggest that you browse the Web sites we list at the end of this chapter for organizations that may provide such support.]*

Chapter 22—Database: SQL, MySQL, DBI and ADO

The vast majority of an organizations' data is stored in databases. In this chapter, we introduce databases as well as the *Structured Query Language (SQL)* for making database queries. The chapter also introduces *MySQL*, an open source, enterprise-level database server, and highlights several key features of this database server. We provide a list of data objects that access MySQL through various programmatic libraries called *database interfaces* (DBIs). We specifically discuss DBIs for Perl, Python and PHP. In addition, a brief discussion of Microsoft's version of data storage, called *universal database access (UDA)*, is provided. A key UDA component is ActiveX Data Objects (ADO), which we introduce in this chapter and use in Chapter 25, Active Server Pages. ADO provides a set of objects used by Microsoft languages such as Visual Basic, Visual C++, C# and Active Server Pages to interact with databases. We list additional resources related to MySQL and Microsoft Access at **www.deitel.com**.

Chapter 23—Wireless[7] Internet Technology

In Chapter 23, we discuss the impact of wireless communications on individuals and businesses. We investigate mobile business applications such as shipping and tracking. We explore location-identification technologies and the services they enable, including wireless marketing and advertising. Privacy issues, related to the ability to locate a user, are carefully examined. The chapter then explores wireless devices and communications technologies and introduces wireless programming. The *Wireless Application Protocol (WAP)* is designed to enable different kinds of wireless devices to communicate and access the Internet using the *Wireless Markup Language* (*WML*). WML tags mark up a Web page to specify how the page

---

7. The reader interested in a deeper treatment of wireless Internet programming may want to consider our book, *Wireless Internet and Mobile Business How to Program*.

is to be formatted on a wireless device. WMLScript helps WAP applications "come alive" by allowing a developer to manipulate WML document content dynamically. In addition to WAP/WML, we explore various platforms and programming languages, such as *Java 2 Micro Edition (J2ME)*, *Qualcomm's Binary Runtime Environment for Wireless (BREW)*, *i-mode*, *Compact HyperText Markup Language* (*cHTML*) and *Bluetooth™ wireless technology.*

Chapter 24—VBScript

JavaScript has become the de facto standard for *client-side scripting*. All major browsers support this language, which has been standardized through the *European Computer Man-ufacturers Association* as *ECMA-262*. *Visual Basic Scripting Edition* (VBScript) is a script-ing language developed by Microsoft. Although it is not supported by many leading browsers, *plug-ins* help some of those browsers understand and process VBScript. VB-Script, however, is the most widely used language for writing Active Server Pages (ASP)—a server-side technology we discuss in Chapters 25 and 26. Chapter 24 prepares you to use VBScript on the client side in Microsoft communities and in Microsoft-based *intranets* (i.e., internal networks that use the same communications protocols as the Internet). It will also prepare you to use VBScript to program Active Server Pages in the next two chapters.

Chapter 25—Active Server Pages (ASP)

This chapter introduces Microsoft's Active Server Pages (ASP), the first of the six server-side software development paradigms we discuss. Active Server Pages can be programmed in a variety of languages—by far the most popular is Microsoft's VBScript (Chapter 24). Active Server Pages implement middle-tier business logic. In this chapter, we introduce the reader to *dynamic content generation* (i.e., the process by which a scripting language gen-erates an XHTML document, an XML document etc.). Chapter examples include a wide range of server-side programming topics, such as writing text files, querying an Access da-tabase and using server-side ActiveX controls to extend Web server functionality. Key ex-amples include an ASP document that allows users to create Web pages, a guestbook application and an ASP document that displays information about the client's browser. This is a crucial chapter for those readers who want to implement Web-based applications by using Microsoft technologies.

Chapter 26—Case Study: Active Server Pages and XML

In this chapter, we build on the material presented in Chapter 25 by creating an online mes-sage forum using ASP. Message forums are "virtual" bulletin boards in which users discuss a variety of topics. The case study presented allows users to post messages to an existing forum and to create new forums. Each forum's data are stored in XML documents that are dynamically manipulated using ASP. This chapter ties together many of the technologies presented earlier in the book, including XHTML, CSS, ASP, XML and XSLT. Chapter ex-ercises ask the reader to modify the case study to delete individual messages from a forum and to delete individual forums.

Chapter 27—Perl and CGI[8] (Common Gateway Interface)

Historically, the most widely used server-side technology for developing Web-based applications has been Perl/CGI. Despite the emergence of newer technologies such as Active Server Pages (Chapters 25 and 26), Python (Chapter 28), PHP (Chapter 29), Java Servlets (Chapter 30) and JavaServer Pages (Chapter 31), the Perl community is well entrenched, and Perl will remain popular for the foreseeable future. Chapter 27 presents an introduction to Perl/CGI, including many real-world, live-code™ examples and discussions, and demonstrations of some of the most recent features of each of these technologies. Key examples demonstrate how to interact with a MySQL database and *regular expressions* (i.e., statements that efficiently search strings for patterns of characters).

Chapter 28—Python[9]

In this chapter, we introduce Python, an interpreted, cross-platform, object-oriented, general-purpose programming language. We begin by presenting basic syntax, data types, control structures and functions. We then introduce *lists* (i.e., data structures similar to a JavaScript array), *tuples* (i.e., immutable lists) and *dictionaries*, which are high-level data structures that store pairs of related data items. String processing and regular expressions are discussed, as is *exception handling*, which provides a structured mechanism for recovering from run-time errors. Chapter examples include implementing an XHTML registration form and showing how to use *cookies* (i.e., small text files written to the client machine). In addition, a three-tier Web-based example queries a MySQL database for author information.

Chapter 29—PHP

In this chapter, we introduce PHP, another popular server-side scripting language for Web-based application development. Similar to Perl and Python, PHP has a large community of users and developers. We begin the chapter by introducing basic syntax, data types, operators and arrays, string processing and regular expressions. Chapter examples include form processing and business logic, connecting to a database and writing cookies. The chapter examples include a three-tier Web-based application that queries a MySQL database.

Chapter 30—Java™ Servlets (Bonus Chapter for Java Developers)

Java servlets represent a fifth popular way of building server-side Web-based applications. Servlets are written in Java (not JavaScript), which requires a substantial book-length treatment to learn. We do not teach Java in *Internet and World Wide Web How to Program: Second Edition*. This chapter (from our book *Advanced Java 2 Platform How to Program*)[10] is provided as a "bonus chapter" for readers familiar with Java. Readers who want to learn Java may want to consider reading our book *Java How to Program, Fourth Edition*.

---

8. The reader interested in a deeper treatment of Perl and CGI may want to consider our book, *Perl How to Program*.
9. The reader interested in a deeper treatment of Python may want to consider our book, *Python How to Program*.
10. This book also discusses Java 2 Enterprise Edition, Java 2 Micro Edition (J2ME), XML, Peer-to-Peer, Java3D, security, Java Database Connectivity (JDBC), Jini and many other advanced topics.

Chapter 31—JavaServer™ Pages (Bonus Chapter for Java Developers)

In this chapter (from our book *Advanced Java 2 Platform How to Program*), we introduce *JavaServer Pages (JSP)*—an extension of Java servlet technology. JavaServer Pages enable Web-application programmers to create dynamic Web content, using familiar XML syntax and scripting with Java. Using JavaServer Pages, Web-application programmers can create *custom tag libraries* that encapsulate complex and dynamic functionality in XML tags. Web-page designers who are not familiar with Java can use these custom tag libraries to integrate information from databases, business-logic components and other resources into dynamically generated Web pages. This chapter is provided as a "bonus" chapter for readers familiar with Java. Readers who want to learn Java may want to consider reading our book *Java How to Program, Fourth Edition*.

Chapter 32—E-Business and E-Commerce

Chapter 32 explores the world of e-business and e-commerce. It begins by discussing the various business models associated with e-businesses. These include storefronts, auctions, portals, dynamic pricing, comparison shopping and demand-sensitive and name-your-price models. We also discuss the management and maintenance of an e-business, which includes advertising and marketing, accepting online payments, securing online transactions and understanding legal issues. We address such topics as branding, e-advertising, customer relationship management, e-wallets, micropayments, privacy and copyright. We also discuss security topics, including public-key cryptography, *Secure Socket Layer (SSL)* and wireless security. The final section in this chapter discusses the emergence of XML and how it enables the standardization of business transactions worldwide.

Chapter 33—Multimedia: Audio, Video, Speech Synthesis and Recognition

This chapter focuses on the explosion of audio, video and speech technology appearing on the Web. We discuss adding sound, video and animated characters to Web pages (primarily using existing audio and video clips). Your first reaction may be a sense of caution, because these are complex technologies about which most readers have had little education. You quickly will see how easy it is to incorporate multimedia into Web pages and control multimedia components with Dynamic HTML. Multimedia files can be large. Some multimedia technologies require that an entire multimedia file be downloaded to the client before the audio or video begins playing. With *streaming audio* and *streaming video* technologies, audio and video can begin playing while the files are downloading, thus reducing delays. Streaming technologies are popular on the Web. This chapter demonstrates how to incorporate the *RealNetworks RealPlayer* into a Web page to receive streaming media. The chapter also includes an extensive set of Internet and Web resources that discuss interesting ways in which designers use multimedia-enhanced Web pages. This chapter introduces an exciting technology called *Microsoft Agent* for adding *interactive animated characters* to an XHTML document. *Agent characters* include *Peedy the Parrot*, *Genie, Merlin* and *Robby the Robot,* as well as those created by third-party developers. Each character allows users to interact with the application, using more natural human communication techniques such as speech. The agent characters accept mouse and keyboard interaction, speak and hear (i.e., they support speech synthesis and speech recognition). With these capabilities, your Web pages can speak to users and can actually respond to their voice commands! Microsoft Agent is included on the CD-ROM that accompanies this book. The chapter exercises ask the reader to create a karaoke machine and to incorporate an agent character into a Web page.

Chapter 34—Accessibility

Currently, the World Wide Web presents a challenge to individuals with disabilities. Multimedia-rich Web sites are difficult for text readers and other programs to interpret, especially for deaf users and users with visual impairments. To rectify this situation, the World Wide Web Consortium (W3C) launched the *Web Accessibility Initiative (WAI)*, which provides guidelines for making Web sites accessible to people with disabilities. This chapter provides a description of these guidelines, such as the use of the **<headers>** tag to make tables more accessible to page readers, the **alt** attribute of the **<img>** tag to describe images, and XHTML and CSS to ensure that a page can be viewed on any type of display or reader. We also introduce *VoiceXML* and *CallXML*, two technologies for increasing the accessibility of Web-based content. VoiceXML helps people with visual impairments to access Web content via speech synthesis and speech recognition. CallXML allows users with visual impairments to access Web-based content through a telephone. In the chapter exercises, readers create their own voicemail applications using CallXML.

Appendix A—XHTML Special Characters

This appendix shows many commonly used XHTML special characters, called *character entity references* by the World Wide Web Consortium (W3C).

Appendix B—Operator Precedence Chart

This appendix contains a JavaScript operator precedence chart.

Appendix C—ASCII Character Set

This appendix contains a table of the 128 ASCII alphanumeric symbols.

Appendix D—Number Systems

This appendix explains the binary, octal, decimal and hexadecimal number systems. It shows how to convert between bases and perform mathematical operations in each base.

Appendix E—XHTML Colors

This appendix explains how to create colors by using either color names or hexadecimal RGB values. Included is a table that matches colors to values.

Appendix F—Career Resources

The Internet presents valuable resources and services for job seekers and employers. Automatic search features allow employees to scan the Web for open positions. Employers also can find job candidates by using the Internet. This greatly reduces the amount of time spent preparing and reviewing resumes, as well as travel expenses for distance recruiting and interviewing. In this chapter, we explore career services on the Web from the perspectives of of job seekers and employers. We introduce comprehensive job sites, industry-specific sites (including sites geared specifically for Java and wireless programmers) and contracting opportunities, as well as additional resources and career services designed to meet the needs of a variety of individuals.

Appendix G—Unicode®

This appendix introduces the *Unicode Standard*, an encoding scheme that assigns unique numeric values to the world's characters. It includes an XML-based example that uses Unicode encoding to print a welcome message in 10 different languages.