1

Introduction

The renaissance of interest in the web that we call Web 2.0 has reached the mainstream.

—Tim O'Reilly

Introduction to Computers and the Internet

The renaissance of interest in the web that we call Web 2.0 has reached the mainstream.
—Tim O'Reilly

Billions of queries stream across the servers of these Internet services—the aggregate thoughtstream of humankind, online.
—John Battelle, *The Search*

People are using the web to build things they have not built or written or drawn or communicated anywhere else.
—Tim Berners-Lee

Some people take what we contribute and extend it and contribute it back [to Ruby on Rails]. That's really the basic open source success story.
—David Heinemeier Hansson, interviewed by Chris Karr at www.Chicagoist.com

OBJECTIVES

In this chapter you will learn:

- Basic computing concepts.
- The different types of programming languages.
- The evolution of the Internet and the World Wide Web.
- What Web 2.0 is and why it's having such an impact among Internet-based and traditional businesses.
- What Rich Internet Applications (RIAs) are and the key software technologies used to build RIAs.

Outline

1.1 Introduction

Welcome to Internet and World Wide Web programming and Web 2.0! And welcome to a walkthrough of the Web 2.0 phenomenon from the technical, business and social perspectives. We've worked hard to create what we hope you'll find to be an informative, entertaining and challenging learning experience. As you read this book, you may want to refer to `www.deitel.com` for updates and additional information.

The technologies you'll learn in this book are fun for novices, and simultaneously are appropriate for experienced professionals who build substantial information systems. *Internet & World Wide Web How to Program, Fourth Edition*, is designed to be an effective learning tool for each of these audiences. How can one book appeal to both groups? The answer is that the core of this book emphasizes achieving program clarity through the proven techniques of structured programming, object-based programming and object-oriented programming. Beginners will learn programming the right way from the beginning. Experienced programmers will find "industrial-strength" code examples. We have attempted to write in a clear and straightforward manner using best practices.

Perhaps most important, the book presents hundreds of working examples and shows the outputs produced when these examples are rendered in browsers or run on computers. We present all concepts in the context of complete working programs. We call this the "live-code approach." These examples are available for download from our website, `www.deitel.com/books/iw3htp4/`.

The early chapters introduce computer fundamentals, the Internet and the web. We show how to use software for browsing the web. We present a carefully paced introduction to "client-side" web programming, using the popular JavaScript language and the closely related technologies of XHTML (Extensible HyperText Markup Language), CSS (Cascading

Style Sheets) and the DOM (Document Object Model). We often refer to "programming" as scripting—for reasons that will soon become clear. Novices will find that the material in the JavaScript chapters presents a solid foundation for the deeper treatment of scripting in the Adobe Flash, Adobe Flex, Microsoft Silverlight, PHP and Ruby on Rails chapters later in the book. Experienced programmers will read the early chapters quickly and find the treatment of scripting in the later chapters to be rigorous and challenging.

Most people are familiar with the exciting things that computers can do. Using this textbook, you'll learn how to command computers to perform specific tasks. **Software** (i.e., the instructions you write to command the computer to perform **actions** and make **decisions**) controls computers (often referred to as **hardware**). JavaScript and PHP are among today's most popular software development languages for web-based applications.

Computer use is increasing in almost every field of endeavor. In an era of steadily rising costs, computing costs have been decreasing dramatically because of rapid developments in both hardware and software technologies. Computers that filled large rooms and cost millions of dollars just two decades ago can now be inscribed on the surfaces of silicon chips smaller than fingernails, costing perhaps a few dollars each. Silicon is one of the most abundant materials on earth—it is an ingredient in common sand. Silicon-chip technology has made computing so economical that more than a billion general-purpose computers worldwide are now helping people in business, industry, government, education and in their personal lives. And billions more computers are embedded in cell phones, appliances, automobiles, security systems, game systems and so much more.

Through the early 1990s most students in introductory programming courses learned only the methodology called structured programming. As you study the various scripting languages in this book, you'll learn both structured programming and the newer methodology called object-based programming. After this, you'll be well prepared to study today's popular full-scale programming languages such as C++, Java, C# and Visual Basic .NET and to learn the even more powerful programming methodology of object-oriented programming. We believe that object-oriented programming will be the key programming methodology for at least several decades.

Today's users are accustomed to applications with rich graphical user interfaces (GUIs), such as those used on Apple's Mac OS X systems, Microsoft Windows systems, various Linux systems and more. Users want applications that employ the multimedia capabilities of graphics, images, animation, audio and video. They want applications that can run on the Internet and the web and communicate with other applications. Users want to apply database technologies for storing and manipulating their business and personal data. They want applications that are not limited to the desktop or even to some local computer network, but that can integrate Internet and web components, and remote databases. Programmers want to use all these capabilities in a truly portable manner so that applications will run without modification on a variety of **platforms** (i.e., different types of computers running different operating systems).

In this book, we present a number of powerful software technologies that will enable you to build these kinds of systems. Early in the book we concentrate on using technologies such as the Extensible HyperText Markup Language (XHTML), JavaScript, CSS, Flash, Flex, Silverlight, Dreamweaver and Extensible Markup Language (XML) to build the portions of web-based applications that reside on the **client side** (i.e., the portions of applications that typically run in your web browsers such as Mozilla's Firefox 2 or

Microsoft's Internet Explorer 7). Later in the book we concentrate on using technologies such as web servers, databases (integrated collections of data), PHP, Ruby on Rails, ASP.NET, ASP.NET Ajax and JavaServer Faces (JSF) to build the server side of web-based applications. These portions of applications typically run on "heavy-duty" computer systems on which organizations' business-critical websites reside. By mastering the technologies in this book, you'll be able to build substantial web-based, client/server, database-intensive, "multitier" applications. We begin with a discussion of computer hardware and software fundamentals. If you are generally familiar with computers, the Internet and the web, you may want to skip some or all of this chapter.

To keep up to date with Internet and web programming developments, and the latest information on *Internet & World Wide Web How to Program, 4/e*, at Deitel & Associates, please register for our free e-mail newsletter, *the Deitel Buzz Online*, at

```
www.deitel.com/newsletter/subscribe.html
```

Please check out our growing list of Internet and web programming, and Internet business Resource Centers at

```
www.deitel.com/resourcecenters.html
```

Each week, we announce our latest Resource Centers in the newsletter. Figure 2 in the Preface includes a complete list of Deitel Resource Centers at the time of this writing. The Resource Centers include links to, and descriptions of, key tutorials, demos, free software tools, articles, e-books, white papers, videos, podcasts, blogs, RSS feeds and more that will help you deepen your knowledge of most of the subjects we discuss in this book.

Errata and updates for the book are posted at

```
www.deitel.com/books/iw3htp4/
```

You're embarking on a challenging and rewarding path. As you proceed, if you have any questions, please send e-mail to

```
deitel@deitel.com
```

We'll respond promptly. We hope that you'll enjoy learning with *Internet & World Wide Web How to Program, Fourth Edition.*

Architecture of Internet & World Wide Web How to Program, 4/e

This book focuses on Web 2.0 and Rich Internet Application (RIA) development. Our goal is to develop webtop applications that have the responsiveness, look and feel of traditional desktop applications. In the interim since the previous edition of this book, Deitel has evolved into a development organization, while maintaining its focus on programming languages textbook and professional book authoring, and corporate training. We're building the infrastructure for the Internet businesses we're designing and developing as part of our Web 2.0 Internet Business Initiative. This edition has been enhanced with discussions of many practical issues we've encountered in developing that infrastructure.

Figure 1.1 shows the architecture of *Internet & World Wide Web How to Program, 4/e.* The book is divided into several parts. The first part, Chapters 1–3, provides an introduction to the Internet and the web, web browsers and Web 2.0. These chapters provide a foundation for understanding Web 2.0 and Rich Internet Application development. Chapter 1 introduces hardware, software, communications and Web 2.0 topics. If you are

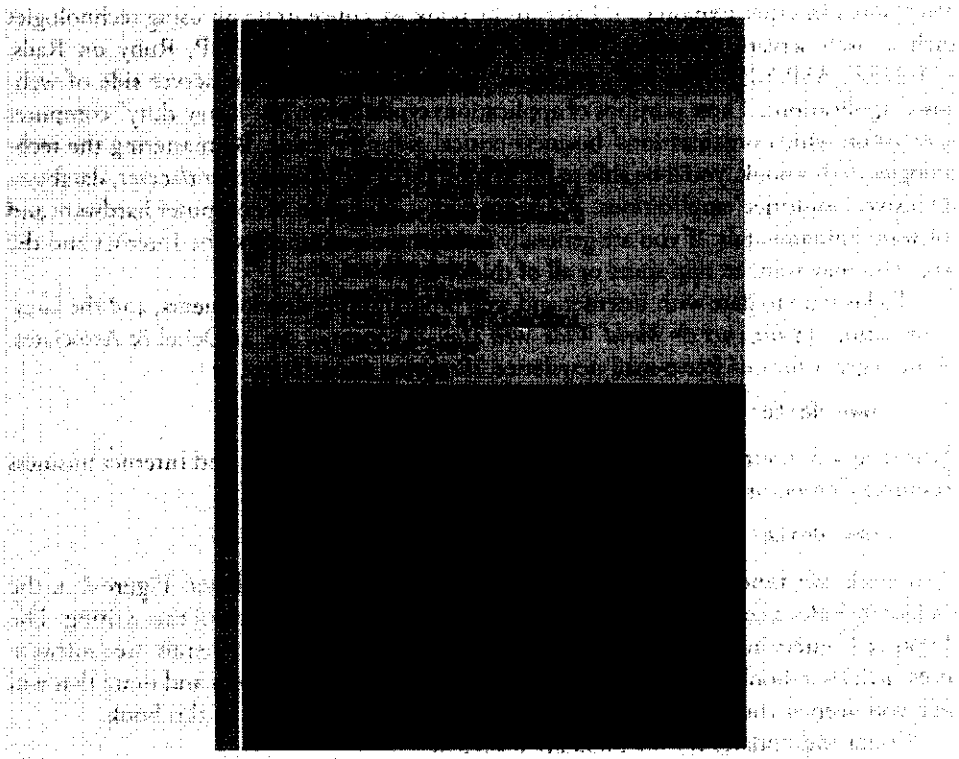

Fig. 1.1 | Architecture of *Internet & World Wide Web How to Program, 4/e.*

a serious web developer, you'll want to test your web applications across many browsers and platforms. The examples for this book target Microsoft's Internet Explorer 7 (IE7) and Mozilla's Firefox 2 (FF2) browsers, each of which is introduced in Chapter 2. The examples execute correctly in both browsers. Many of the examples will also work in other browsers such as Opera and Safari. Many of the examples will not work on earlier browsers. Microsoft Windows users should upgrade to IE7 and install FF2; readers with other operating systems should install Firefox 2. Web browsers—a crucial component of web applications—are free, as are most of the software technologies we present in this book. Chapter 3 discusses Web 2.0 from technical, business and social perspectives.

The second part of the book, Chapters 4–15, presents a 12-chapter treatment of Ajax component technologies that concludes with Chapter 15's treatment of Ajax development. Ajax is not a new technology—we've been writing about all but one of its component technologies since the first edition of this book in 1999, and many of the technologies existed before that. However, Ajax is one of the key technologies of Web 2.0 and RIAs. Several later chapters in the book demonstrate technologies that encapsulate Ajax functionality to help make it operate across a wide variety of browsers and browser versions.

The third part of the book, Chapters 15–28, focuses on both the client and server sides of the GUI and graphical part of RIA development. Here we cover client-side technologies such as Adobe Flash, Adobe Flex and Microsoft Silverlight that use, or can be combined with, Ajax or Ajax-like capabilities to develop RIAs. Each of these technologies

also can consume web services. Next, we present the server side of web application development with discussions of web servers (IIS and Apache), databases, several server-side scripting languages such as PHP and Ruby on Rails, and several server-side frameworks such as ASP.NET 2.0 and JavaServer Faces. We complete our server-side discussion with a chapter on building web services.

You may have noticed that Chapter 15, Ajax-Enabled Rich Internet Applications, overlaps the second and third parts of the book. Chapter 15 serves as a bridge from "raw" Ajax development to sophisticated RIA development.

1.2 What Is a Computer?

A **computer** is a device that can perform computations and make logical decisions billions of times faster than human beings can. For example, many of today's personal computers can perform several billion additions per second. A person could operate a desk calculator for an entire lifetime and still not complete as many calculations as a powerful personal computer can perform in one second! (Points to ponder: How would you know whether the person added the numbers correctly? How would you know whether the computer added the numbers correctly?) Today's fastest **supercomputers** can perform trillions of additions per second!

Computers process data under the control of sets of instructions called **computer programs**. These programs guide the computer through orderly sets of actions specified by people called **computer programmers**.

A computer consists of various devices referred to as hardware (e.g., the keyboard, screen, mouse, hard disk, memory, DVD drives and processing units). The programs that run on a computer are referred to as software. Hardware costs have been declining dramatically in recent years, to the point that personal computers have become a commodity. In this book, you'll learn proven methods that are reducing software development costs—object-oriented programming and object-oriented design.

1.3 Computer Organization

Regardless of differences in physical appearance, virtually every computer may be envisioned as divided into six **logical units** or sections:

1. **Input unit.** This is the "receiving" section of the computer. It obtains information (data and computer programs) from **input devices** and places this information at the disposal of the other units for processing. Most information is entered into computers through keyboards and mouse devices. Information also can be entered in many other ways, including by speaking to your computer, scanning images, uploading digital photos and videos, and receiving information from a network, such as the Internet.
2. **Output unit.** This is the "shipping" section of the computer. It takes information that the computer has processed and places it on various **output devices** to make the information available for use outside the computer. Most information output from computers today is displayed on screens, printed on paper or used to control other devices. Computers also can output their information to networks, such as the Internet.

3. **Memory unit.** This is the rapid-access, relatively low-capacity "warehouse" section of the computer. It stores computer programs while they are being executed. It retains information that has been entered through the input unit, so that it will be immediately available for processing when needed. The memory unit also retains processed information until it can be placed on output devices by the output unit. Information in the memory unit is typically lost when the computer's power is turned off. The memory unit is often called either **memory** or **primary memory**.
4. **Arithmetic and logic unit (ALU).** This is the "manufacturing" section of the computer. It is responsible for performing calculations, such as addition, subtraction, multiplication and division. It contains the decision mechanisms that allow the computer, for example, to compare two items from the memory unit to determine whether they are equal, or if one is larger than the other.
5. **Central processing unit (CPU).** This is the computer's "administrative" section. It coordinates and supervises the other sections' operations. The CPU tells the input unit when information should be read into the memory unit, tells the ALU when information from the memory unit should be used in calculations and tells the output unit when to send information from the memory unit to certain output devices. Many of today's computers have multiple CPUs and, hence, can perform many operations simultaneously—such computers are called **multiprocessors**.
6. **Secondary storage unit.** This is the computer's long-term, high-capacity "warehousing" section. Programs or data not actively being used by the other units normally are placed on secondary storage devices, such as your hard drive, until they are needed, possibly hours, days, months or even years later. Information in secondary storage takes much longer to access than information in primary memory, but the cost per unit of secondary storage is much less than that of primary memory. Other secondary storage devices include CDs and DVDs, which can hold hundreds of millions and billions of characters, respectively.

1.4 Machine Languages, Assembly Languages and High-Level Languages

Programmers write instructions in various programming languages, some directly understandable by computers and others requiring intermediate **translation** steps. Hundreds of computer languages are in use today. These may be divided into three general types:

1. Machine languages
2. Assembly languages
3. High-level languages

Any computer can directly understand only its own **machine language**. Machine language is the "natural language" of a computer and as such is defined by its hardware design. [*Note:* Machine language is often referred to as **object code**. This term predates "object-oriented programming." These two uses of "object" are unrelated.] Machine languages generally consist of strings of numbers (ultimately reduced to 1s and 0s) that instruct computers to perform their most elementary operations one at a time. Machine languages are **machine dependent** (i.e., a particular machine language can be used on only one type of computer). Such languages are cumbersome for humans, as illustrated by the following

section of an early machine-language program that adds overtime pay to base pay and stores the result in gross pay:

```
+1300042774
+1400593419
+1200274027
```

Machine-language programming was simply too slow, tedious and error prone for most programmers. Instead of using the strings of numbers that computers could directly understand, programmers began using English-like abbreviations to represent elementary operations. These abbreviations formed the basis of **assembly languages**. **Translator programs** called **assemblers** were developed to convert early assembly-language programs to machine language at computer speeds. The following section of an assembly-language program also adds overtime pay to base pay and stores the result in gross pay:

```
load    basepay
add     overpay
store   grosspay
```

Although such code is clearer to humans, it is incomprehensible to computers until translated to machine language.

Computer usage increased rapidly with the advent of assembly languages, but programmers still had to use many instructions to accomplish even the simplest tasks. To speed the programming process, **high-level languages** were developed in which single statements could be written to accomplish substantial tasks. Translator programs called **compilers** convert high-level language programs into machine language. High-level languages allow programmers to write instructions that look almost like everyday English and contain commonly used mathematical notations. A payroll program written in a high-level language might contain a statement such as

```
grossPay = basePay + overTimePay;
```

From your standpoint, obviously, high-level languages are preferable to machine and assembly language. C, C++, Microsoft's .NET languages (e.g., Visual Basic, Visual C++ and Visual C#) and Java are among the most widely used high-level programming languages. All of the Internet and web application development languages that you'll learn in this book are high-level languages.

The process of compiling a high-level language program into machine language can take a considerable amount of computer time. **Interpreter** programs were developed to execute high-level language programs directly, although much more slowly. In this book, we study several key programming languages, including JavaScript, ActionScript, PHP and Ruby on Rails—each of these **scripting languages** is processed by interpreters. We also study markup languages such as XHTML and XML, which can be processed by interpreted scripting languages. You'll see that interpreters have played an especially important role in helping scripting languages and markup languages achieve their goal of portability across a variety of platforms.

Performance Tip 1.1

Interpreters have an advantage over compilers in scripting. An interpreted program can begin executing as soon as it is downloaded to the client's machine, without the need to be compiled before it can execute. On the downside, scripts generally run much slower than compiled code.

Portability Tip 1.1

Interpreted languages are more portable than compiled languages. Interpreters can be implemented for each platform on which the interpreted languages need to execute.

Software Engineering Observation 1.1

Interpreted languages are more dynamic than compiled languages. For example, server-side applications can generate code in response to user interactions, and that code can then be interpreted in a browser.

1.5 History of the Internet and World Wide Web

In the late 1960s, one of the authors (HMD) was a graduate student at MIT. His research at MIT's Project MAC (now the Laboratory for Computer Science—the home of the World Wide Web Consortium) was funded by ARPA—the Advanced Research Projects Agency of the Department of Defense. ARPA sponsored a conference at which several dozen ARPA-funded graduate students were brought together at the University of Illinois at Urbana-Champaign to meet and share ideas. During this conference, ARPA rolled out the blueprints for networking the main computer systems of about a dozen ARPA-funded universities and research institutions. They were to be connected with communications lines operating at a then-stunning 56 Kbps (i.e., 56,000 bits per second)—this at a time when most people (of the few who could) were connecting over telephone lines to computers at a rate of 110 bits per second. There was great excitement at the conference. Researchers at Harvard talked about communicating with the Univac 1108 "supercomputer" at the University of Utah to handle calculations related to their computer graphics research. Many other intriguing possibilities were raised. Academic research about to take a giant leap forward. Shortly after this conference, ARPA proceeded to implement the **ARPANET**, which eventually evolved into today's **Internet**.

Things worked out differently from what was originally planned. Rather than enabling researchers to share each other's computers, it rapidly became clear that enabling researchers to communicate quickly and easily via what became known as **electronic mail** (**e-mail**, for short) was the key early benefit of the ARPANET. This is true even today on the Internet, as e-mail facilitates communications of all kinds among a billion people worldwide.

One of the primary goals for ARPANET was to allow multiple users to send and receive information simultaneously over the same communications paths (e.g., phone lines). The network operated with a technique called **packet switching**, in which digital data was sent in small bundles called **packets**. The packets contained address, error-control and sequencing information. The address information allowed packets to be routed to their destinations. The sequencing information helped in reassembling the packets—which, because of complex routing mechanisms, could actually arrive out of order—into their original order for presentation to the recipient. Packets from different senders were intermixed on the same lines. This packet-switching technique greatly reduced transmission costs, as compared with the cost of dedicated communications lines.

The network was designed to operate without centralized control. If a portion of the network failed, the remaining working portions would still route packets from senders to receivers over alternative paths for reliability.

The protocol for communicating over the ARPANET became known as TCP—the **Transmission Control Protocol**. TCP ensured that messages were properly routed from sender to receiver and that they arrived intact.

As the Internet evolved, organizations worldwide were implementing their own networks for both intraorganization (i.e., within the organization) and interorganization (i.e., between organizations) communications. A wide variety of networking hardware and software appeared. One challenge was to get these different networks to communicate. ARPA accomplished this with the development of IP—the **Internet Protocol**, truly creating a **"network of networks,"** the current architecture of the Internet. The combined set of protocols is now commonly called TCP/IP.

Initially, Internet use was limited to universities and research institutions; then the military began using the Internet. Eventually, the government decided to allow access to the Internet for commercial purposes. Initially, there was resentment in the research and military communities—these groups were concerned that response times would become poor as "the Net" became saturated with users.

In fact, the exact opposite has occurred. Businesses rapidly realized that they could tune their operations and offer new and better services to their clients, so they started spending vast amounts of money to develop and enhance the Internet. This generated fierce competition among communications carriers and hardware and software suppliers to meet this demand. The result is that **bandwidth** (i.e., the information-carrying capacity) on the Internet has increased tremendously and costs have decreased significantly.

The **World Wide Web** allows computer users to locate and view multimedia-based documents on almost any subject over the Internet. Though the Internet was developed decades ago, the web is a relatively recent creation. In 1989, **Tim Berners-Lee** of CERN (the European Organization for Nuclear Research) began to develop a technology for sharing information via hyperlinked text documents. Berners-Lee called his invention the **HyperText Markup Language** (HTML). He also wrote communication protocols to form the backbone of his new information system, which he called the World Wide Web. In particular, he wrote the **Hypertext Transfer Protocol** (HTTP)—a communications protocol used to send information over the web. Web use exploded with the availability in 1993 of the Mosaic browser, which featured a user-friendly graphical interface. Marc Andreessen, whose team at NCSA developed Mosaic, went on to found Netscape, the company that many people credit with initiating the explosive Internet economy of the late 1990s.

In the past, most computer applications ran on computers that were not connected to one another, whereas today's applications can be written to communicate among the world's computers. The Internet mixes computing and communications technologies. It makes our work easier. It makes information instantly and conveniently accessible worldwide. It enables individuals and small businesses to get worldwide exposure. It is changing the way business is done. People can search for the best prices on virtually any product or service. Special-interest communities can stay in touch with one another. Researchers can be made instantly aware of the latest breakthroughs. The Internet and the web are surely among humankind's most profound creations.

1.6 World Wide Web Consortium (W3C)

In October 1994, Tim Berners-Lee founded an organization—called the **World Wide Web Consortium** (W3C)—devoted to developing nonproprietary, interoperable technologies

for the World Wide Web. One of the W3C's primary goals is to make the web universally accessible—regardless of ability, language or culture. The W3C home page (`www.w3.org`) provides extensive resources on Internet and web technologies.

The W3C is also a standardization organization. Web technologies standardized by the W3C are called **Recommendations**. W3C Recommendations include the Extensible HyperText Markup Language (XHTML), Cascading Style Sheets (CSS), HyperText Markup Language (HTML—now considered a "legacy" technology) and the Extensible Markup Language (XML). A recommendation is not an actual software product, but a document that specifies a technology's role, syntax rules and so forth.

1.7 Web 2.0

In 2003 there was a noticeable shift in how people and businesses were using the web and developing web-based applications. The term **Web 2.0** was coined by **Dale Dougherty** of **O'Reilly Media**[1] in 2003 to describe this trend. Although it became a major media buzzword, few people really know what Web 2.0 means. Generally, Web 2.0 companies use the web as a platform to create collaborative, community-based sites (e.g., social networking sites, blogs, wikis, etc.).

Web 1.0 (the state of the web through the 1990s and early 2000s) was focused on a relatively small number of companies and advertisers producing content for users to access (some people called it the "brochure web"). Web 2.0 *involves* the user—not only is the content often created by the users, but users help organize it, share it, remix it, critique it, update it, etc. One way to look at Web 1.0 is as a *lecture*, a small number of professors informing a large audience of students. In comparison, Web 2.0 is a *conversation*, with everyone having the opportunity to speak and share views.

Web 2.0 is providing new opportunities and connecting people and content in unique ways. Web 2.0 embraces an **architecture of participation**—a design that encourages user interaction and community contributions. You, the user, are the most important aspect of Web 2.0—so important, in fact, that in 2006, *TIME Magazine*'s "Person of the Year" was "you."[2] The article recognized the social phenomenon of Web 2.0—the shift away from a powerful few to an empowered many. Several popular blogs now compete with traditional media powerhouses, and many Web 2.0 companies are built almost entirely on user-generated content. For websites like MySpace, Facebook, Flickr™, YouTube, eBay and Wikipedia, users create the content, while the companies provide the platforms. These companies *trust their users*—without such trust, users cannot make significant contributions to the sites.

The architecture of participation has influenced software development as well. Open source software is available for anyone to use and modify with few or no restrictions. Using **collective intelligence**—the concept that a large diverse group of people will create smart ideas—communities collaborate to develop software that many people believe is better and more robust than proprietary software. Rich Internet Applications (RIAs) are being

1. O'Reilly, T. "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software." September 2005 <`http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1`>.
2. Grossman, L. "TIME's Person of the Year: You." *TIME*, December 2006 <`http://www.time.com/time/magazine/article/0,9171,1569514,00.html`>.

developed using technologies (such as Ajax) that have the look and feel of desktop software, enhancing a user's overall experience. Software as a Service (SaaS)—software that runs on a server instead of a local computer—has also gained prominence because of sophisticated new technologies and increased broadband Internet access.

Search engines, including Google™, Yahoo!, MSN, Ask™, and many more, have become essential to sorting through the massive amount of content on the web. Social bookmarking sites such as del.icio.us and Ma.gnolia allow users to share their favorite sites with others. Social media sites such as Digg™, Spotplex™ and Netscape enable the community to decide which news articles are the most significant. The way we find the information on these sites is also changing—people are **tagging** (i.e., labeling) web content by subject or keyword in a way that helps anyone locate information more effectively.

Web services have emerged and, in the process, have inspired the creation of many Web 2.0 businesses. Web services allow you to incorporate functionality from existing applications and websites into your own web applications quickly and easily. For example, using Amazon Web Services™, you can create a specialty bookstore and earn revenues through the Amazon Associates Program; or, using Google™ Maps web services with eBay web services, you can build location-based "mashup" applications to find auction items in certain geographical areas. Web services, inexpensive computers, abundant high-speed Internet access, open source software and many other elements have inspired new, exciting, **lightweight business models** that people can launch with only a small investment. Some types of websites with rich and robust functionality that might have required hundreds of thousands or even millions of dollars to build in the 1990s can now be built for nominal amounts of money.

In the future, we'll see computers learn to understand the meaning of the data on the web—the beginnings of the Semantic Web are already appearing. Continual improvements in hardware, software and communications technologies will enable exciting new types of applications.

These topics and more are covered in a detailed walkthrough in Chapter 3, Dive Into Web 2.0. The chapter highlights the major characteristics and technologies of Web 2.0, providing examples of popular Web 2.0 companies and Web 2.0 Internet business and monetization models. You'll learn about user-generated content, blogging, content networks, social networking, location-based services and more. In Chapters 4–28, you'll learn key software technologies for building web-based applications in general, and Ajax-enabled, web-based Rich Internet Applications in particular. See our Web 2.0 Resource Center at `www.deitel.com/web2.0/` for more information.

1.8 Personal, Distributed and Client/Server Computing

In 1977, Apple Computer popularized **personal computing**. Computers became so economical that people could buy them for their own personal or business use. In 1981, IBM, the world's largest computer vendor, introduced the IBM Personal Computer. This quickly legitimized personal computing in business, industry and government organizations, where IBM mainframes were heavily used.

These computers were for the most part "stand-alone" units—people transported disks back and forth between them to share information (this was often called "sneakernet"). Although early personal computers were not powerful enough to timeshare several users, these machines could be linked together in computer networks, sometimes over

telephone lines and sometimes in **local area networks (LANs)** within an organization. This led to the phenomenon of **distributed computing**, in which an organization's computing, instead of being performed only at some central computer installation, is distributed over networks to the sites where the organization's work is performed. Personal computers were powerful enough to handle the computing requirements of individual users as well as the basic communications tasks of passing information between computers electronically.

Today's personal computers are as powerful as the million-dollar machines of just a few decades ago. The most powerful desktop machines—called **workstations**—provide individual users with enormous capabilities. Information is shared easily across computer networks, where computers called **servers** (file servers, database servers, web servers, etc.) offer data storage and other capabilities that may be used by **client** computers distributed throughout the network, hence the term **client/server computing**. Today's popular operating systems, such as UNIX, Linux, Mac OS X and Microsoft's Windows-based systems, provide the kinds of capabilities discussed in this section.

1.9 Hardware Trends

The Internet community thrives on the continuing stream of dramatic improvements in hardware, software and communications technologies. In general, people expect to pay at least a little more for most products and services every year. The opposite generally has been the case in the computer and communications industries, especially with regard to the hardware costs of supporting these technologies. For many decades, and with no change expected in the foreseeable future, hardware costs have fallen rapidly. This is a phenomenon of technology. **Moore's Law** states that the power of hardware doubles every two years, while the price remains essentially the same.[3] Significant improvements also have occurred in the communications field, especially in recent years, with the enormous demand for communications bandwidth attracting tremendous competition, forcing communications bandwidth to increase and prices to decline. We know of no other fields in which technology moves so quickly and costs fall so rapidly.

When computer use exploded in the 1960s and 1970s, there was talk of the huge improvements in human productivity that computing and communications would bring about. However, these productivity improvements did not immediately materialize. Organizations were spending vast sums on computers and distributing them to their workforces, but without immediate productivity gains. On the hardware side, it was the invention of microprocessor chip technology and its wide deployment in the late 1970s and 1980s which laid the groundwork for significant productivity improvements in the 1990s. On the software side, productivity improvements are now coming from object technology, which we use throughout this book.

Recently, hardware has been moving toward mobile, wireless technology. Small handheld devices are now more powerful than early 1970s supercomputers. Portability is now a major focus for the computer industry. Wireless data-transfer speeds have become so fast that many Internet users' primary web access is through wireless networks. The next few years will see dramatic advances in wireless capabilities for personal users and businesses.

---

3. Moore, G. "Cramming More Components onto Integrated Circuits." *Electronics*, April 1965 <ftp://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf>.

1.10 Key Software Trend: Object Technology

One of the authors, HMD, remembers the great frustration felt in the 1960s by software development organizations, especially those working on large-scale projects. During his undergraduate years, he had the privilege of working summers at a leading computer vendor on the teams developing timesharing, virtual-memory operating systems. This was a great experience for a college student. But, in the summer of 1967, reality set in when the company "decommitted" from producing as a commercial product the particular system on which hundreds of people had been working for many years. It was difficult to get this thing called software right—software is "complex stuff."

Improvements to software technology did emerge, with the benefits of structured programming (and the related disciplines of **structured systems analysis and design**) being realized in the 1970s. Not until the technology of object-oriented programming became widely used in the 1990s, though, did software developers feel they had the necessary tools for making major strides in the software development process.

What are objects and why are they special? Actually, object technology is a packaging scheme that helps us create meaningful software units. These can be large and are highly focused on particular applications areas. There are date objects, time objects, paycheck objects, invoice objects, audio objects, video objects, file objects, record objects and so on. In fact, almost any noun can be reasonably represented as an object.

We live in a world of objects. Just look around you. There are cars, planes, people, animals, buildings, traffic lights, elevators and the like. Before object-oriented languages appeared, procedural programming languages (such as Fortran, COBOL, Pascal, BASIC and C) were focused on actions (verbs) rather than on things or objects (nouns). Programmers living in a world of objects programmed primarily using verbs. This made it awkward to write programs. Now, with the availability of popular object-oriented languages, such as C++, Java, Visual Basic and C#, programmers continue to live in an object-oriented world *and* can program in an object-oriented manner. This is a more natural process than procedural programming and has resulted in significant productivity gains.

A key problem with procedural programming is that the program units do not effectively mirror real-world entities, so these units are not particularly reusable. It's not unusual for programmers to "start fresh" on each new project and have to write similar software "from scratch." This wastes time and money, as people repeatedly "reinvent the wheel." With object technology, the software entities created (called **classes**), if properly designed, tend to be reusable on future projects. Using libraries of reusable componentry can greatly reduce effort required to implement certain kinds of systems (compared to the effort that would be required to reinvent these capabilities on new projects).

Software Engineering Observation 1.2

Extensive class libraries of reusable software components are available on the Internet. Many of these libraries are free.

Software Engineering Observation 1.3

Some organizations report that the key benefit object-oriented programming gives them is not software that is reusable but, rather, software that is more understandable, better organized and easier to maintain, modify and debug. This can be significant, because perhaps as much as 80 percent of software cost is associated not with the original efforts to develop the software, but with the continued evolution and maintenance of that software throughout its lifetime.

1.11 JavaScript: Object-Based Scripting for the Web

JavaScript is an object-based scripting language with strong support for proper software engineering techniques. Students learn to create and manipulate objects from the start in JavaScript. JavaScript is available free in today's popular web browsers.

Does JavaScript provide the solid foundation of programming principles typically taught in first programming courses—a portion of the intended audience for this book? We think so.

The JavaScript chapters of the book are more than just an introduction to the language. They also present an introduction to computer programming fundamentals, including control structures, functions, arrays, recursion, strings and objects. Experienced programmers will read Chapters 6–13 quickly and master JavaScript by reading our live-code examples and by examining the corresponding screenshots. Beginners will learn computer programming in these carefully paced chapters by reading the code explanations and completing the many exercises.

JavaScript is a powerful scripting language. Experienced programmers sometimes take pride in creating strange, contorted, convoluted JavaScript code. This kind of coding makes programs more difficult to read, test and debug. This book is also geared for novice programmers; for all readers we stress program clarity.

Good Programming Practice 1.1

Write your programs in a simple and straightforward manner. This is sometimes referred to as KIS ("keep it simple"). One key aspect of keeping it simple is another interpretation of KIS—"keep it small." Do not "stretch" the language by trying bizarre uses.

You'll see that JavaScript is a portable scripting language and that programs written in JavaScript can run in many web browsers. Actually, portability is an elusive goal.

Portability Tip 1.2

Although it is easier to write portable programs in JavaScript than in many other programming languages, differences among interpreters and browsers make portability difficult to achieve. Simply writing programs in JavaScript does not guarantee portability. Programmers occasionally need to research platform variations and write their code accordingly.

Portability Tip 1.3

When writing JavaScript programs, you need to deal directly with cross-browser portability issues. Such issues are hidden by JavaScript libraries (e.g., Dojo, Prototype, Script.aculo.us and ASP.NET Ajax) which provide powerful, ready-to-use capabilities that simplify JavaScript coding by making it cross-browser compatible.

Error-Prevention Tip 1.1

Always test your JavaScript programs on all systems and in all web browsers for which they are intended.

Good Programming Practice 1.2

Read the documentation for the JavaScript version you are using to access JavaScript's rich collection of features.

Error-Prevention Tip 1.2

Your computer and JavaScript interpreter are good teachers. If you are not sure how a feature works, even after studying the documentation, experiment and see what happens. Study each error or warning message and adjust the code accordingly.

JavaScript was created by Netscape, the company that created the first widely successful web browser. Both Netscape and Microsoft have been instrumental in the standardization of JavaScript by ECMA International (formerly the European Computer Manufacturers Association) as ECMAScript. In Chapters 16–17, we discuss Adobe Flash, which uses another scripting language named ActionScript. ActionScript and JavaScript are converging in the next version of the JavaScript standard (JavaScript 2/ECMA Script version 4) currently under development by ECMA. This will result in a universal client scripting language, greatly simplifying web application development.

1.12 Browser Portability

Ensuring a consistent look and feel on client-side browsers is one of the great challenges of developing web-based applications. Currently, a standard does not exist to which software developers must adhere when creating web browsers. Although browsers share a common set of features, each browser might render pages differently. Browsers are available in many versions and on many different platforms (Microsoft Windows, Apple Macintosh, Linux, UNIX, etc.). Vendors add features to each new version that sometimes result in cross-platform incompatibility issues. Clearly it is difficult to develop web pages that render correctly on all versions of each browser. In this book we develop web applications that execute on both the Internet Explorer 7 and Firefox 2 browsers.

Portability Tip 1.4

The web is populated with many different browsers, which makes it difficult for authors and web application developers to create universal solutions. The W3C is working toward the goal of a universal client-side platform.

1.13 C, C++ and Java

C

The C language was developed by Dennis Ritchie at Bell Laboratories. C was implemented in 1972. C initially became known as the development language of the UNIX operating system. Today, virtually all new major operating systems are written in C and/or C++.

C++

Bjarne Stroustrup developed C++, an extension of C, in the early 1980s. C++ provides a number of features that "spruce up" the C language, but more importantly, it provides capabilities for object-oriented programming. C++ is a hybrid language: It is possible to program in either a C-like style (procedural programming), in which the focus is on actions, or an object-oriented style, in which the focus is on objects, or both. C and C++ have influenced many subsequent programming languages, such as Java, C#, JavaScript and PHP, each of which has a syntax similar to C and C++.

Java

Microprocessors are having a profound impact in intelligent consumer electronic devices. Recognizing this, Sun Microsystems in 1991 funded an internal corporate research project code-named Green to provide software for these devices. The project resulted in the development of a C++-based language that its creator, James Gosling, called Oak after an oak tree outside his window at Sun. It was later discovered that there already was a computer language called Oak. When a group of Sun people visited a local coffee shop, the name **Java** was suggested and it stuck.

The Green project ran into some difficulties. The marketplace for intelligent consumer electronic devices did not develop in the early 1990s as quickly as Sun had anticipated. The project was in danger of being canceled. By sheer good fortune, the World Wide Web exploded in popularity in 1993, and Sun saw the immediate potential of using Java to add **dynamic content** (e.g., interactivity, animations and the like) to web pages. This breathed new life into the project.

Sun formally announced Java at an industry conference in May 1995. Java garnered the attention of the business community because of the phenomenal interest in the web. Java is now used to develop large-scale enterprise applications, to enhance the functionality of web servers (the computers that provide the content we see in our web browsers), to provide applications for consumer devices (e.g., cell phones, pagers and personal digital assistants) and for many other purposes.

1.14 BASIC, Visual Basic, Visual C++, C# and .NET

The **BASIC** (Beginner's All-purpose Symbolic Instruction Code) programming language was developed in the mid-1960s at Dartmouth College as a means of writing simple programs. BASIC's primary purpose was to familiarize novices with programming techniques. Microsoft's Visual Basic language, introduced in the early 1990s to simplify the development of Microsoft Windows applications, has become one of the most popular programming languages in the world.

Microsoft's latest development tools are part of its corporatewide strategy for integrating the Internet and the web into computer applications. This strategy is implemented in Microsoft's **.NET platform**, which provides the capabilities developers need to create computer applications that can execute on computers distributed across the Internet. Microsoft's three primary programming languages are **Visual Basic** (based on the original BASIC), **Visual C++** (based on C++) and **Visual C#** (a relatively new language based on C++ and Java that was developed expressly for the .NET platform). Developers using .NET can write software components in the language they are most familiar with, then form applications by combining those components with others written in any .NET language.

1.15 Software Technologies

In this section, we discuss some software engineering topics and buzzwords that you'll hear in the software development community. We've created Resource Centers on most of these topics, with many more on the way.

Agile Software Development is a set of methodologies that try to get software implemented quickly with fewer resources than previous methodologies. Check out the Agile Alliance (`www.agilealliance.org`) and the Agile Manifesto (`www.agilemanifesto.org`).

Refactoring involves reworking code to make it clearer and easier to maintain while preserving its functionality. It's widely employed with agile development methodologies. Many refactoring tools are available to do major portions of the reworking automatically.

Design patterns are proven architectures for constructing flexible and maintainable object-oriented software. The field of design patterns tries to enumerate those recurring patterns, encouraging software designers to reuse them to develop better-quality software with less time, money and effort.

Game programming. The computer game business is larger than the first-run movie business. College courses and even majors are now devoted to the sophisticated software techniques used in game programming. Chapter 17 discusses building interactive games with Adobe Flash CS3. Also check out our Resource Centers on Game Programming, C++ Game Programming and Programming Projects.

Open source software is developed in a way unlike the proprietary development that dominated software's early years and remains strong today. With open source development, individuals and companies contribute their efforts in developing, maintaining and evolving software in exchange for the right to use that software for their own purposes, typically at no charge. Open source code generally gets scrutinized by a much larger audience than proprietary software, so bugs may be removed faster. Open source also encourages more innovation. Sun recently open sourced Java. Some organizations you'll hear a lot about in the open source community are the Eclipse Foundation (the Eclipse IDE is popular for C++ and Java software development), the Mozilla Foundation (the creators of the Firefox browser), the Apache Software Foundation (the creators of the Apache web server) and SourceForge (which provides the tools for managing open source projects and currently has over 150,000 open source projects under development).

Linux is an open source operating system and one of the greatest successes of the open source movement. Apache is the most popular open source web server. **MySQL** (see Chapters 22–24) is an open source database management system. **PHP** (see Chapter 23) is the most popular open source server-side "scripting" language for developing Internet-based applications. **LAMP** is an acronym for the set of open source technologies that many developers used to build web applications—it stands for Linux, Apache, MySQL and PHP (or Perl or Python—two other scripting languages used for similar purposes).

Ruby on Rails (see Chapter 24) combines the scripting language Ruby with the Rails web application framework developed by the company 37Signals. Their book, *Getting Real*, is a must read for today's web application developers; read it free at `gettingreal.37signals.com/toc.php`. Many Ruby on Rails developers have reported significant productivity gains over using other languages when developing database-intensive web applications.

Software has generally been viewed as a product; most software still is offered this way. If you want to run an application, you buy a software package from a software vendor. You then install that software on your computer and run it as needed. As new versions of the software appear, you upgrade your software, often at significant expense. This process can become cumbersome for organizations with tens of thousands of systems that must be maintained on a diverse array of computer equipment. With **Software as a Service (SaaS)**, the software runs on servers elsewhere on the Internet. When those servers are updated, all clients worldwide see the new capabilities; no local installation is needed. You access the service through a browser—these are quite portable, so you can run the same applications

on different kinds of computers from anywhere in the world. Salesforce.com, Google, Microsoft and 37Signals all offer SaaS.

1.16 Notes about *Internet & World Wide Web How to Program, 4/e*

In 1995, we saw an explosion of interest in the Internet and the World Wide Web. We immersed ourselves in these technologies, and a clear picture started to emerge in our minds of the next direction to take in writing textbooks for introductory programming courses. **Electronic commerce**, or **e-commerce**, as it is typically called, began to dominate the business, financial and computer industry news. This was a reconceptualization of the way business should be conducted. We still wanted to teach programming principles, but we felt compelled to do it in the context of the technologies that businesses and organizations need to create Internet-based and web-based applications. With this realization, the first edition of *Internet & World Wide Web How to Program* was born and published in December of 1999.

Internet & World Wide Web How to Program, Fourth Edition teaches programming languages and programming language principles. In addition, we focus on the broad range of technologies that will help you build real-world Internet-based and web-based applications that interact with other applications and with databases. These capabilities allow you to develop the kinds of enterprise-level, distributed applications popular in industry today.

You'll learn computer programming and basic principles of computer science and information technology. You also will learn proven software development methods—top-down stepwise-refinement, functionalization and object-based programming. Our primary programming language is JavaScript, a compact language that is especially designed for developing Internet- and web-based applications. Chapters 6–13 present a rich discussion of JavaScript and its capabilities, including dozens of complete examples followed by screen images that illustrate typical program inputs and outputs.

After you learn programming principles from the detailed JavaScript discussions, we present condensed treatments of four other popular Internet/web programming languages for building the server side of Internet- and web-based client/server applications. Chapter 23 introduces the popular PHP scripting language. Chapter 24 introduces Ruby, the scripting language used with the Ruby on Rails framework for rapid development of database-driven web applications. In Chapter 25, we discuss ASP.NET 2.0—Microsoft's technology for server-side scripting. ASP.NET pages can be written in Visual Basic and C#; we code ASP.NET pages using Visual Basic. In Chapters 26–27, we discuss JavaServer Faces, which uses the Java programming language. Finally, in Chapter 28, we discuss web services (using examples in both Java and ASP.NET).

1.17 Web Resources

`www.deitel.com/`

Check this site frequently for updates, corrections and additional resources for all Deitel & Associates, Inc., publications.

`www.deitel.com/resourcecenters.html`

Check out the complete list of Deitel Resource Centers, including numerous programming, open source, Web 2.0 and Internet business topics.

`netforbeginners.about.com`
The About.com *Internet for Beginners* guide provides valuable resources for further exploration of the history and workings of the Internet and the web.

`www.learnthenet.com/english/index.html`
Learn the Net is a website containing a complete overview of the Internet, the web and the underlying technologies. The site contains much information appropriate for novices.

`www.w3.org`
The World Wide Web Consortium (W3C) website offers a comprehensive description of web technologies. For each Internet technology with which the W3C is involved, the site provides a description of the technology, its benefits to web designers, the history of the technology and the future goals of the W3C in developing the technology.

Summary

Section 1.1 Introduction

- In an era of steadily rising costs, computing costs have been decreasing dramatically because of rapid developments in both hardware and software technologies.
- Technologies such as Extensible HyperText Markup Language (XHTML), JavaScript, Flash, Flex, Dreamweaver and Extensible Markup Language (XML) are used to build the portions of web-based applications that reside on the client side (i.e., the portions of applications that typically run on web browsers such as Firefox or Microsoft's Internet Explorer).
- Technologies such as web servers, databases, ASP.NET, PHP, Ruby on Rails and JavaServer Faces are used to build the server side of web-based applications. These parts of applications typically run on "heavy-duty" computer systems on which organizations' business-critical websites reside.

Section 1.2 What Is a Computer?

- A computer is a device capable of performing computations and making logical decisions at speeds billions of times faster than human beings can.
- A computer processes data under the control of sets of instructions called computer programs, which guide it through orderly sets of actions specified by computer programmers.
- The various devices that comprise a computer system are referred to as hardware.
- The computer programs that run on a computer are referred to as software.

Section 1.3 Computer Organization

- The input unit is the "receiving" section of the computer. It obtains information from input devices and places it at the disposal of the other units for processing.
- The output unit is the "shipping" section of the computer. It takes information processed by the computer and places it on output devices to make it available for use outside the computer.
- The memory unit is the rapid-access, relatively low-capacity "warehouse" section of the computer. It retains information that has been entered through the input unit, making it immediately available for processing when needed, and retains information that has already been processed until it can be placed on output devices by the output unit.
- The arithmetic and logic unit (ALU) is the "manufacturing" section of the computer. It is responsible for performing calculations and making decisions.
- The central processing unit (CPU) is the "administrative" section of the computer. It coordinates and supervises the operation of the other sections.

- The secondary storage unit is the long-term, high-capacity "warehousing" section of the computer. Programs or data not being used by the other units are normally placed on secondary storage devices (e.g., disks) until they are needed, possibly hours, days, months or even years later.

Section 1.4 Machine Languages, Assembly Languages and High-Level Languages

- Any computer can directly understand only its own machine language, which generally consists of strings of numbers ultimately reduced to 1s and 0s that instruct the computer to perform its most elementary operations.
- English-like abbreviations form the basis of assembly languages. Translator programs called assemblers convert assembly-language programs to machine language.
- Compilers translate high-level language programs into machine-language programs. High-level languages contain English words and conventional mathematical notations.
- Interpreter programs directly execute high-level language programs, eliminating the need to compile them into machine language.

Section 1.5 History of the Internet and World Wide Web

- In the late 1960s, ARPA, the Advanced Research Projects Agency of the U.S. Department of Defense rolled out the blueprints for networking the main computer systems of about a dozen ARPA-funded universities and research institutions. ARPA then proceeded to implement the ARPANET, the predecessor to today's Internet.
- The World Wide Web allows computer users to locate and view multimedia-based documents (i.e., documents with text, graphics, animations, audios or videos) on almost any subject.
- In 1989, Tim Berners-Lee of CERN began to develop the World Wide Web and several communication protocols that form the backbone of the web.
- Web use exploded with the availability in 1993 of the Mosaic browser, which featured a user-friendly graphical interface. Marc Andreessen, whose team at NCSA developed Mosaic, went on to found Netscape, the company that many people credit with initiating the explosive Internet economy of the late 1990s.

Section 1.6 World Wide Web Consortium (W3C)

- In October 1994, Tim Berners-Lee founded the World Wide Web Consortium (W3C)—an organization devoted to developing nonproprietary, interoperable technologies for the web.

Section 1.7 Web 2.0

- Web 2.0 companies use the web as a platform to create collaborative, community-based sites (e.g., social networking sites, blogs, wikis, etc.).
- Web 1.0 (the state of the web through the 1990s and early 2000s) was focused on a relatively small number of companies and advertisers producing content for users to access.
- Web 2.0 embraces an architecture of participation—a design that encourages user interaction and community contributions.
- Using the collective intelligence—the concept that a large diverse group of people will create smart ideas—communities collaborate to develop open source software that many people believe is better and more robust than proprietary software.
- Rich Internet Applications (RIAs) are being developed using technologies (such as Ajax) that have the look and feel of desktop software, enhancing a user's overall experience.
- Web services, inexpensive computers, abundant high-speed Internet access, open source software and many other elements have inspired new, exciting, lightweight business models that people can launch with only a small investment.

Section 1.8 Personal, Distributed and Client/Server Computing

- Apple Computer popularized personal computing.
- IBM's Personal Computer quickly legitimized personal computing in business, industry and government organizations, where IBM mainframes were heavily used.
- Although early personal computers were not powerful enough to timeshare several users, these machines could be linked together in computer networks, sometimes over telephone lines and sometimes in local area networks (LANs) within an organization. This led to the phenomenon of distributed computing.
- Today's personal computers are as powerful as the million-dollar machines of just a few decades ago, and information is shared easily across computer networks.

Section 1.9 Hardware Trends

- Moore's Law states that the power of hardware doubles every two years, while the price remains essentially the same.

Section 1.10 Key Software Trend: Object Technology

- Objects are essentially reusable software components that model real-world items.
- Not until object-oriented programming became widely used in the 1990s did software developers feel they had the tools to make major strides in the software development process.
- Object technology is a packaging scheme that helps us create meaningful software units.
- A key problem with procedural programming is that the program units do not effectively mirror real-world entities, so these units are not particularly reusable.
- With object technology, the software entities created (called classes), if properly designed, tend to be reusable on future projects. Using libraries of reusable componentry can greatly reduce effort required to implement certain kinds of systems.
- Some organizations report that the key benefit object-oriented programming gives them is the production of software which is more understandable, better organized and easier to maintain, modify and debug.

Section 1.11 JavaScript: Object-Based Scripting for the Web

- JavaScript is an object-based scripting language with strong support for proper software engineering techniques.
- JavaScript was created by Netscape. Both Netscape and Microsoft have been instrumental in the standardization of JavaScript by ECMA International as ECMAScript.

Section 1.12 Browser Portability

- Ensuring a consistent look and feel on client-side browsers is one of the great challenges of developing web-based applications.

Section 1.13 C, C++ and Java

- C initially became known as the development language of the UNIX operating system. Today, virtually all new major operating systems are written in C and/or C++.
- C++ provides a number of features that "spruce up" the C language, but more importantly, it provides capabilities for object-oriented programming.
- Java is used to create dynamic and interactive content for web pages, develop enterprise applications, enhance web-server functionality, provide applications for consumer devices and more.

Section 1.14 BASIC, Visual Basic, Visual C++, C# and .NET

- The BASIC programming language was developed in the mid-1960s at Dartmouth College. Its primary purpose was to familiarize novices with programming techniques.
- Microsoft's Visual Basic was introduced in the early 1990s to simplify the process of developing Microsoft Windows applications.
- Microsoft has a corporatewide strategy for integrating the Internet and the web into computer applications. This strategy is implemented in Microsoft's .NET platform.
- The .NET platform's three primary programming languages are Visual Basic, Visual C++ and Visual C#.
- .NET developers can write software components in their preferred language, then form applications by combining those components with components written in any .NET language.

Section 1.15 Software Technologies

- Agile Software Development is a set of methodologies that try to get software implemented quickly with fewer resources than previous methodologies.
- Refactoring involves reworking code to make it clearer and easier to maintain while preserving its functionality.
- Design patterns are proven architectures for constructing flexible and maintainable object-oriented software.
- Open source development allows individuals and companies to contribute their efforts in developing, maintaining and evolving software in exchange for the right to use that software for their own purposes, typically at no charge.
- With Software as a Service (SaaS), the software runs on servers elsewhere on the Internet, rather than on the desktop.

Terminology

actions
Agile Software Development
architecture of participation
arithmetic and logic unit (ALU)
ARPANET
assemblers
assembly language
bandwidth
BASIC
C
C++
central processing unit (CPU)
class
client side
client/server computing
collective intelligence
compilers
computer
computer program
computer programmer
CSS
Dale Dougherty
data
data structure
decision
design pattern
distributed computing
DOM (Document Object Model)
dynamic content
electronic commerce (e-commerce)
electronic mail (e-mail)
function
game programming
hardware
high-level languages
HTML (HyperText Markup Language)

HTTP (Hypertext Transfer Protocol)
input device
input unit
Internet
interpreter
IP (Internet Protocol)
Java
JavaScript
LAMP
library
lightweight business models
Linux
local area networks (LANs)
logical unit
machine dependent
machine language
memory
memory unit
method
Moore's Law
multiprocessor
MySQL
.NET platform
O'Reilly Media
object code
object-based programming
object-oriented programming
open source software
output devices
output unit
packet
packet switching
personal computer
PHP
platform
primary memory
refactoring
Ruby on Rails
scripting
scripting language
secondary storage unit
server side
server
software
Software as a Service (SaaS)
structured programming
structured systems analysis and design
supercomputer
tagging
TCP (Transmission Control Protocol)
TCP/IP
translation
translator program
Visual Basic
Visual C#
Visual C++
Web 1.0
Web 2.0
workstation
World Wide Web
World Wide Web Consortium (W3C)
XHTML (Extensible HyperText Markup Language)
XML (Extensible Markup Language)

Self-Review Exercises

1.1 Fill in the blanks in each of the following:

a) The company that popularized personal computing was _____.

b) The computer that made personal computing legitimate in business and industry was the _____.

c) Computers process data under the control of sets of instructions called computer _____.

d) The six key logical units of the computer are the _____, _____, _____, _____, _____ and the _____.

e) The three classes of languages discussed in the chapter are _____, _____, and _____.

f) The programs that translate high-level language programs into machine language are called _____.
g) _____, or labeling content, is another key part of the collaborative theme of Web 2.0.
h) With Internet applications, the desktop evolves to the _____.
i) _____ involves reworking code to make it clearer and easier to maintain while preserving its functionality.
j) With _____ development, individuals and companies contribute their efforts in developing, maintaining and evolving software in exchange for the right to use that software for their own purposes, typically at no charge.
k) The _____ was the predecessor to the Internet.
l) The information-carrying capacity of a communications medium like the Internet is called _____.
m) The acronym TCP/IP stands for _____.

1.2 Fill in the blanks in each of the following statements.
a) The _____ allows computer users to locate and view multimedia-based documents on almost any subject over the Internet.
b) _____ founded an organization—called the World Wide Web Consortium (W3C)—devoted to developing nonproprietary, interoperable technologies for the World Wide Web.
c) _____ are reusable software components that model items in the real world.
d) In a typical client/server relationship, the _____ requests that some action be performed and the _____ performs the action and responds.

Exercises

1.3 Categorize each of the following items as either hardware or software:
a) CPU
b) ALU
c) input unit
d) an editor program

1.4 Fill in the blanks in each of the following statements:
a) Which logical unit of the computer receives information from outside the computer for use by the computer? _____.
b) The process of instructing the computer to solve specific problems is called _____.
c) What type of computer language uses English-like abbreviations for machine-language instructions? _____.
d) Which logical unit of the computer sends information that has already been processed by the computer to various devices so that the information may be used outside the computer? _____.
e) Which logical units of the computer retain information? _____.
f) Which logical unit of the computer performs calculations? _____.
g) Which logical unit of the computer makes logical decisions? _____.
h) The level of computer language most convenient for you to write programs quickly and easily is _____.
i) The only language that a computer directly understands is called that computer's _____.
j) Which logical unit of the computer coordinates the activities of all the other logical units? _____.

2

Web Browser Basics: Internet Explorer and Firefox

Give us the tools, and we will finish the job.
—Sir Winston Churchill

We must learn to explore all the options and possibilities that confront us in a complex and rapidly changing world.
—J. William Fulbright

OBJECTIVES

In this chapter you will learn:

- To understand the Microsoft Internet Explorer 7 (IE7) and Mozilla Firefox 2 (FF2) web browsers' capabilities.
- To use IE7 and FF2 to search the information available on the World Wide Web.
- To customize a browser according to your own needs and preferences.
- To understand the differences among various browsers.

Outline

2.1 Introduction to the Internet Explorer 7 and Firefox 2 Web Browsers

The Internet is an essential medium for communicating and interacting with people worldwide. The need to publish and share information has fueled the rapid growth of the web. **Web browsers** are software programs that allow users to access the web's rich content. Whether for business or personal use, millions of people use web browsers to access the tremendous amount of information available on the web and to share or exchange this content with other users. The www portion of the Internet, which we encounter often in this chapter, is made up of hyperlinked documents written in XHTML and rich media.

Popular web browsers at the time of publication are Microsoft's *Internet Explorer*, Mozilla's *Firefox*, Apple's *Safari* and Opera Software's *Opera*. This chapter focuses on the features of Internet Explorer (IE7) and Firefox 2 (FF2), which are the most widely used of these browsers. All examples in this book are supported by both IE7 and FF2.

2.2 Connecting to the Internet

A computer alone is not enough to access the Internet. In addition to web browser software, the computer needs specific hardware and a connection to an Internet Service Provider to view web pages. This section describes the components that enable Internet access.

First, a computer must have a **modem** or **network card**. A modem is hardware that enables a computer to connect to a **network** via phone lines. A modem converts data to audio tones and transmits the data over phone lines. A network card, also called a **network interface card** (NIC), is hardware that allows a computer to connect to the Internet through a network or a high-speed Internet connection, such as a **local area network** (LAN), **cable modem** or **Digital Subscriber Line** (DSL).

After ensuring that a computer has a modem or a network card (most computers come with one or both of these), the next step is to register with an **Internet Service Provider** (ISP). Computers connect to an ISP using a modem and phone line, or via a NIC using a LAN, DSL or cable modem. The ISP connects computers to the Internet. Most college and university campuses offer network connections, and many communities now offer wireless access. If a network connection is not available, then popular commercial ISPs,

such as AOL (www.aol.com), Comcast (www.comcast.net), Earthlink (www.earthlink.net), Verizon (www.verizon.com), Microsoft Network (www.msn.com) and NetZero (www.netzero.net) are alternatives.

Bandwidth and cost are two considerations when deciding which commercial ISP service to use. Bandwidth refers to the amount of data that can be transferred through a communications medium in a fixed amount of time. Different ISPs offer different types of high-speed connections, called **broadband connections**—which include DSL, cable modem and **Integrated Services Digital Network (ISDN)**—and slower **dial-up connections**. Each connection type has a different bandwidth and cost to users.

Broadband is a category of high-bandwidth Internet service that is most often provided to home users by cable television and telephone companies. DSL is a broadband service that allows computers to be connected at all times to the Internet over existing phone lines, without interfering with telephone services. DSL requires a special modem provided by the ISP. Like DSL, cable modems enable the computer to be connected to the Internet at all times. Cable modems transmit data over the cables that bring television to homes and businesses. Unlike DSL, the bandwidth is shared by many users. This sharing can reduce the bandwidth available to each person when many use the system simultaneously. ISDN provides Internet service over either digital or standard telephone lines. ISDN requires specialized hardware, called a **terminal adapter (TA)**, which is usually obtained from the ISP.

Dial-up service uses an existing telephone line. If a computer is connected to the Internet, the user usually cannot receive voice calls during this time. If the voice calls do connect, the Internet connection is interrupted. To prevent this, users may choose to install an extra phone line dedicated to Internet service.

Fiber optics are replacing traditional metal cables in many computer networks due to their greater bandwidth and mechanical advantages that provide a better signal. Though their popularity is currently limited by the high cost of materials and installation, consistent improvements in the industry will allow fiber optic cables to become a key element of the communications industry in the near future.

Once a computer is connected to a network, the user must choose a web browser for navigating the Internet. Internet Explorer is preinstalled on all Windows machines, and your version can be updated at www.microsoft.com/ie. Firefox can be downloaded at www.mozilla.com/firefox, and can be installed on many different operating systems. When installing this browser, select **Custom** when prompted for a setup type, and ensure that the **DOM Inspector** option is selected in the next screen. Doing so will ensure that you have additional Firefox functionality that we discuss in Chapter 12.

2.3 Internet Explorer 7 and Firefox 2 Features

A web browser is software that allows the user to view certain types of Internet files in an interactive environment. Figure 2.1 shows the Deitel Home Page using Internet Explorer 7 web browser, and Fig. 2.2 uses Firefox 2. The **URL (Uniform Resource Locator)** http://www.deitel.com is found in the **Address** bar in IE7, and the **Location** bar in FF2. The URL specifies the address (i.e., location) of the web page displayed in the browser window. Each web page on the Internet is associated with a unique URL. URLs usually begin with http://, which stands for **Hypertext Transfer Protocol (HTTP)**, the standard protocol (or set of communication rules) for transferring web documents over the Internet. URLs of websites that handle private information, such as credit card numbers, often

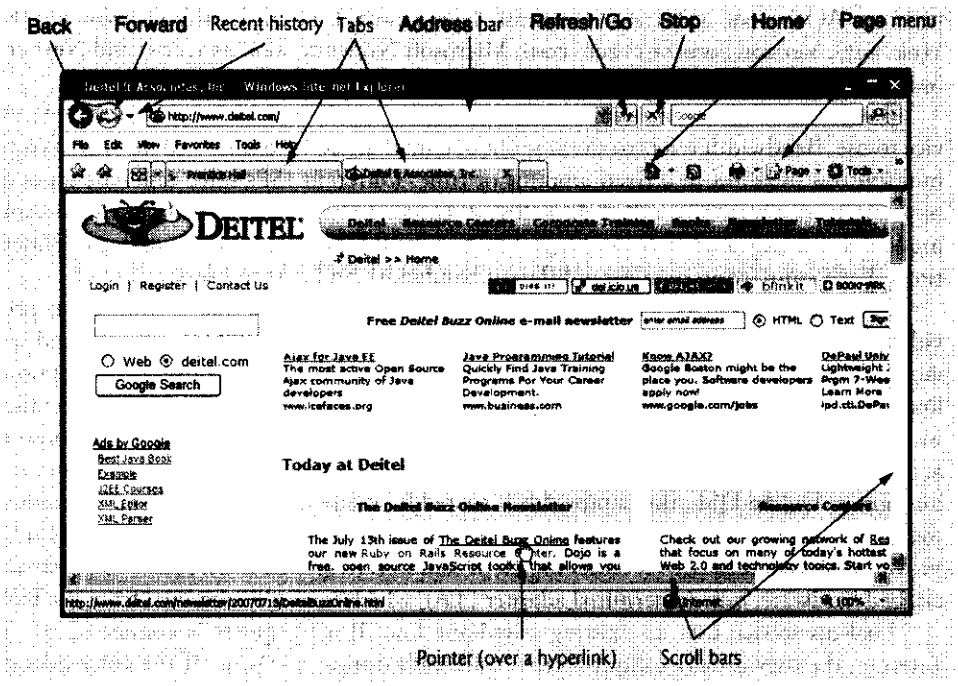

Fig. 2.1 | Deitel website in Internet Explorer 7.

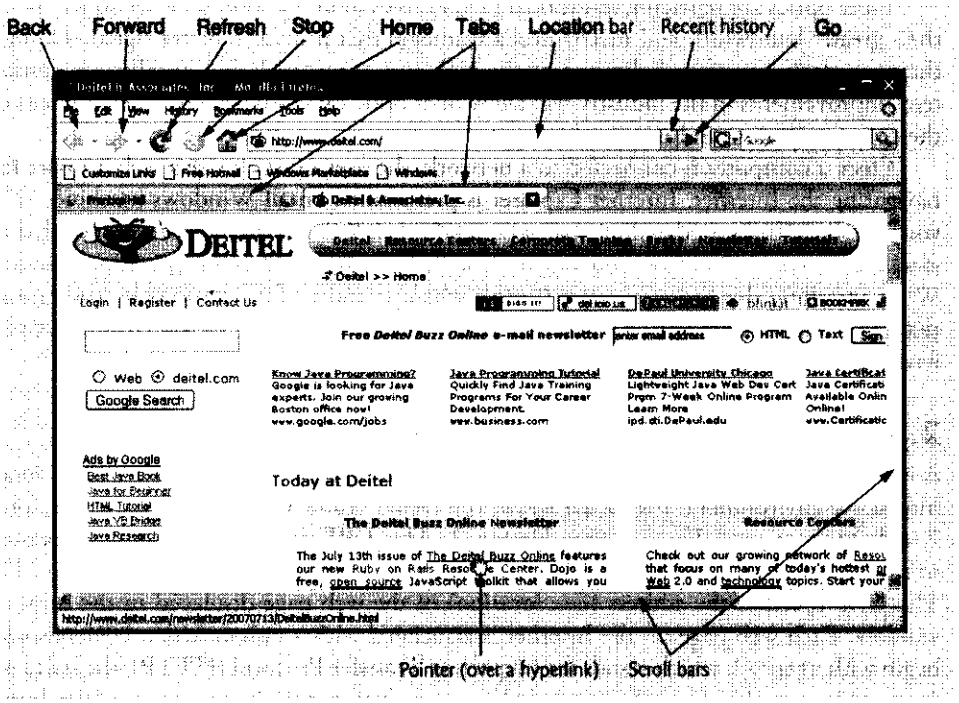

Fig. 2.2 | Deitel website in Firefox 2.

begin with `https://`, the abbreviation for **Hypertext Transfer Protocol over Secure Sockets Layer** (HTTPS), the standard for transferring encrypted data on the web.

There are several techniques for navigating between URLs. You can click the **Address** field and type a web page's URL, then press *Enter* or click **Go** (in IE7, this is the same button as **Refresh**) to request the web page located at that URL. For example, to visit Yahoo!'s website, type `www.yahoo.com` in the **Address** bar and press the *Enter* key. Clicking **Refresh** loads the latest version of the web page from the current website. IE7 and FF2, as well as most other popular browsers, add the `http://` prefix to the website name because HTTP is the default protocol used for the web.

Hyperlinks

Another way to navigate the web is via visual elements on web pages called **hyperlinks** that, when clicked, load a specified web document. Both images and text may be hyperlinked. When the mouse pointer hovers over a hyperlink, the default arrow pointer changes into a hand with the index finger pointing upward. Often hyperlinked text appears underlined and as in different color from regular text in a web page. Originally used as a publishing tool for scientific research, hyperlinks are widely used to reference sources, or sites that have more information on a particular topic. The paths created by hyperlinking create the effect of the "web."

Hyperlinks can reference other web pages, e-mail addresses, files and more. If a hyperlink's URL is in the form `mailto:`*emailAddress*, clicking the link loads your default e-mail program and opens a **message window** addressed to the specified e-mail address. Note that these hyperlinks are generally displayed on the screen as just the e-mail address or the recipient's name.

If a hyperlink references a file that the browser is incapable of displaying, the browser prepares to **download** the file, and generally prompts the user for information about how the file should be stored. When a file is downloaded, it is copied onto the user's computer. Programs, documents, images and sound files are all examples of downloadable files.

Tabbed Browsing

Many browsers, including IE7 and FF2, provide **tabbed browsing**. Holding down the *Ctrl* key and pressing the letter *T* while in the IE7 or FF2 browser opens another tab in the same window, allowing the user to browse multiple pages without cluttering the desktop with many windows. [*Note:* For Mac users, all references to the *Ctrl* key in this chapter's shortcuts should be replaced with the *Command* key.] Also, pressing *Ctrl* while clicking a link will open the requested page in a new tab. Clicking on the tabs switches between the different pages in the browser, and web pages are then accessed normally. Using tabs is an excellent way to keep the browser organized when viewing multiple pages at once.

Using the History Feature

IE7 and FF2 maintain a **History** list of previously visited URLs in chronological order. This feature allows users to return to recently visited websites easily. The history feature can be accessed several different ways. The simplest and most frequently used method is to click the **Forward** and **Back** buttons located at the top of the browser window (see Fig. 2.1). The **Back** button reloads into the browser the page you last visited. Assuming that you used the **Back** button to view previously visited pages, the **Forward** button would load the next URL from the history into the browser. The keyboard shortcut for **Forward** is *<Alt>* and

the *Right Arrow* key or just *Shift* and *Backspace*, and the shortcut for **Back** is *<Alt>* and the *Left Arrow* key or simply *Backspace*.

In IE7, the user can view the last and next nine web pages visited and the current page by clicking the down arrows immediately to the right of the **Forward** button; the user can then request one of the recently viewed pages by clicking the title of the page in the drop-down list. In FF2, there are separate menus to the right of both the **Forward** and the **Back** buttons. Each displays the previous and following fifteen pages in the history, respectively. Note that these methods only display history results from the browser's current **session**, which is the period when the browser remains open. In IE7 and FF2, there is a menu to the right of the address bar which displays a longer but more basic history of visited sites (it does not include any URLs accessed through hyperlinks), including websites that were visited in previous sessions. Another way to display sites from a previous session is to use **History**.

Selecting **History** from the down-arrow menu in IE7, or clicking the **History** menu, then the **Show In Sidebar** option in FF2, divides the browser window into two sections: the **History** window (on the left) and the content window (Figs. 2.3–2.4). In IE7, clicking the yellow star icon in the upper left of the window, then selecting the **History** option, displays a similar menu. By default, the **History** window lists the URLs visited in the past twenty days in IE7 and nine days in FF2.

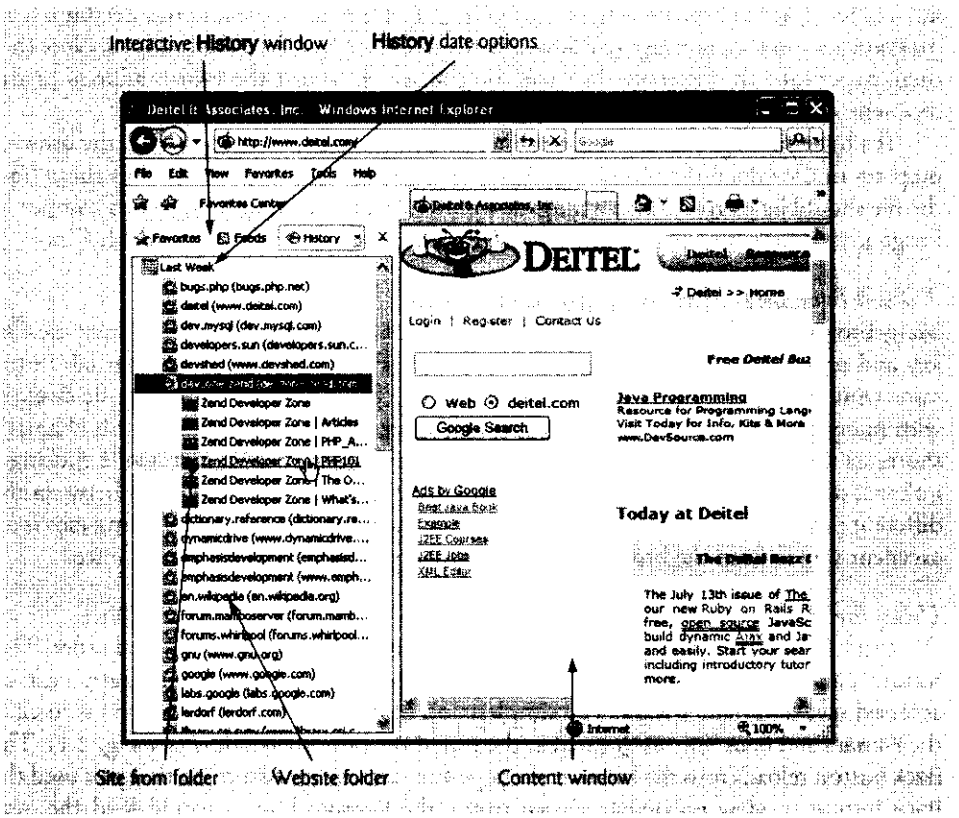

Fig. 2.3 | The **History** menu in Internet Explorer 7.

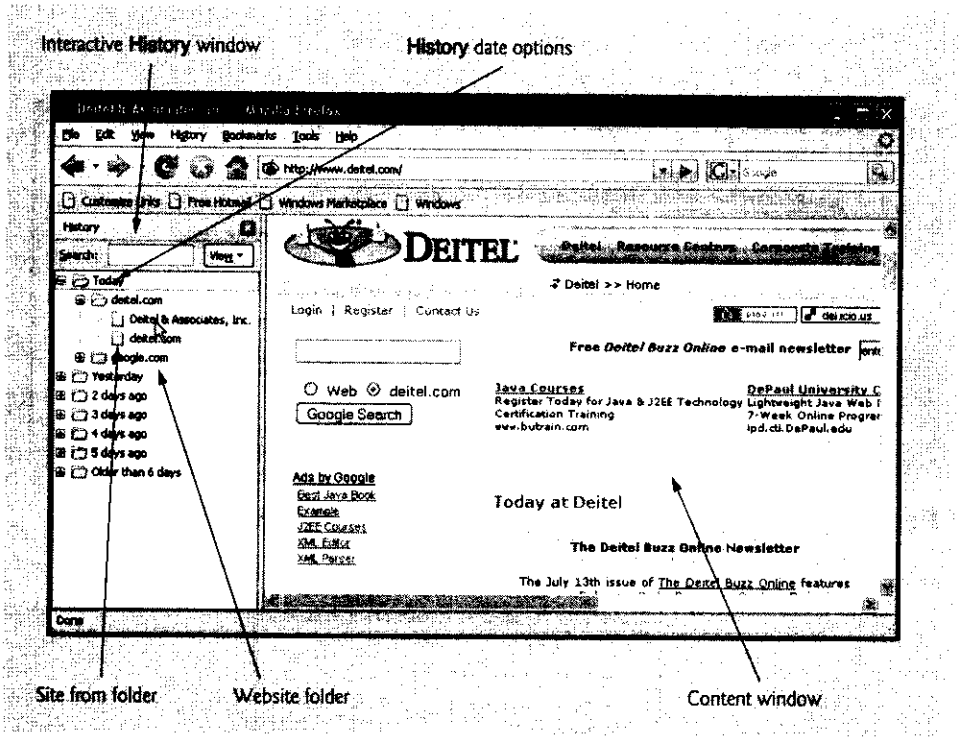

Fig. 2.4 | The **History** menu in Firefox 2.

The **History** window contains heading levels ordered chronologically. Within each time frame (e.g., **Today**) headings are alphabetized by website name (although the organization can be changed clicking the **History** drop-down menu in IE7 or the **View** drop-down menu of FF2, both located in the **History** window). This window is useful for finding previously visited websites without having to remember the exact URL. Selecting a URL from the **History** window loads the web page into the content window.

AutoComplete

URLs from the history can be displayed in a drop-down list when a user types a URL into the **Address** bar. This feature is called **AutoComplete**. Any URL from this drop-down list can be selected with the mouse to load the web page at that URL into the browser (Fig. 2.5).

Off-Line Browsing

For some users, such as those with dial-up connections, maintaining a connection for long periods of time may not be practical. For this reason, web pages can be saved directly to the computer's hard drive for **off-line** browsing (i.e., browsing while not connected to the Internet). Select **Save As...** in IE7, or **Save Page As...** in FF2, both from the **File** menu to save a web page and all its components, including the images. [*Note:* To display the **File** menu in IE7, press the *Alt* key.] This option is also available under the **Page** menu in IE7 (Fig. 2.1). Individual images from a website can also be saved by clicking the image with the right mouse button and selecting **Save Picture As...** (IE7) or **Save Image As...** (FF2) from the displayed **context menu** (Fig. 2.6).

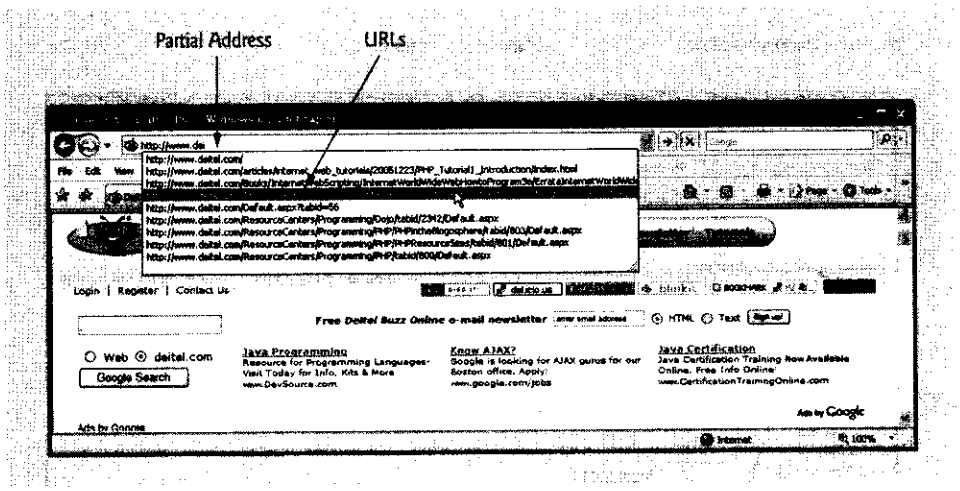

Fig. 2.5 | AutoComplete suggests possible URLs when given a partial address.

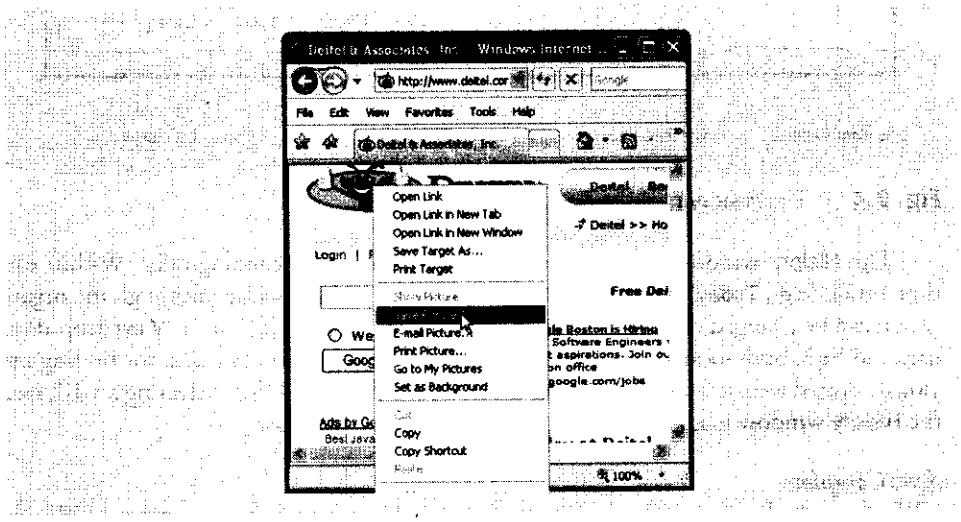

Fig. 2.6 | Saving a picture from a website.

Downloads

As mentioned earlier, files from the Internet may be copied to a computer's hard drive by a process called **downloading**. This section discusses the types of documents commonly downloaded from the Internet and techniques for downloading them. [*Note:* You should always be cautious when downloading files from the Internet, as they may contain viruses. Only download from sites that you trust.]

Some common Internet downloads are **applications** (i.e., software that performs specific functions, such as word processing), **plug-ins** and **extensions**. Plug-ins are specialized pieces of software that help the browser support additional content types. An example of an IE7 and FF2 plug-in is the **Acrobat Reader** from **Adobe, Inc.** (`www.adobe.com/products/acrobat/readstep2.html`), which allows users to view PDF (Portable

Document Format) documents that otherwise cannot be rendered by the browser. Another popular plug-in allows the browser to render **Flash** content, which adds audio, video and animation effects to a website. To view sites enabled with Flash, download the Adobe Flash Player plug-in at `www.adobe.com/products/flashplayer`. Microsoft's rich media plug-in, **Silverlight**, is available for download at `silverlight.net/GetStarted`. (Both Flash and Silverlight are discussed in much greater depth in Chapters 16, 17 and 19). Normally the browser prompts the user to download a plug-in when one is needed. Plug-ins may also be downloaded from CNET (`www.download.com`). This site has a large, searchable index and database of many plug-in programs available for download.

Extensions are add-ons that enhance the preexisting functionality of the browser. Examples of extensions include blog editors, universal uploaders and various translation dictionaries and tools. Many IE7 add-ons can be found at `www.ieaddons.com`, and FF2 add-ons can be browsed and downloaded at `https://addons.mozilla.org`.

Viewing Source Code

Clicking on the **View** menu followed by the **Source** option in IE7 and **Page Source** in FF2 allows you to view the **source code**, or the original code written to create the web page you are viewing. Generally, source code is easy for humans to read and interpret, and allows the viewer to understand how the programmer created the page. For example, if an element of a web page does not display properly, examining the source code can help to inform the user what the programmer was trying to do. Examining source code is a useful tool for debugging your own code, or for learning how web developers create some of the elements you see on the web.

2.4 Customizing Browser Settings

Browsers have many settings that determine how sites are displayed, how security measures are applied and how outputs are rendered. Most of these settings are located in the **Internet Options** dialog (Fig. 2.7) in the **Tools** menu of IE7, and in **Options** under the **Tools** menu in FF2 in Windows (Fig. 2.8) [*Note:* For Firefox on a Mac, this is called the **Preferences** menu.]. The default settings are usually adequate for normal browsing, but these settings can be customized to suit each user's preferences.

Some privacy settings for IE7 and FF2 can be set under the **Privacy** tab. In IE7 there are six levels of privacy. The most lenient level permits the downloading of **cookies** (text files that are placed on the computer by websites to retain or gather information about the user); the most strict level blocks all cookies from all websites and constantly updates a report to the user about browsing privacy. Using this level may prevent certain websites from working correctly. In FF2 the **Privacy** tab displays options about how data is remembered in the system and when cookies should be accepted.

Security options for both browsers can be found under the **Security** tab. The browsers' options are significantly different, but both allow you to specify how much information you want to hide from unfamiliar sites, as well as how much of the site's content you would like to block from your own computer. Both browsers allow you to distinguish between trusted sites and the rest of the web, and to browse safe sites with lower security settings.

A personal home page can be specified under the **General** tab in IE7 and **Main** in FF2. The home page is the web page that loads when the browser is first opened and appears when the **Home** button at the top of the browser window is clicked (Figs. 2.1–2.2).

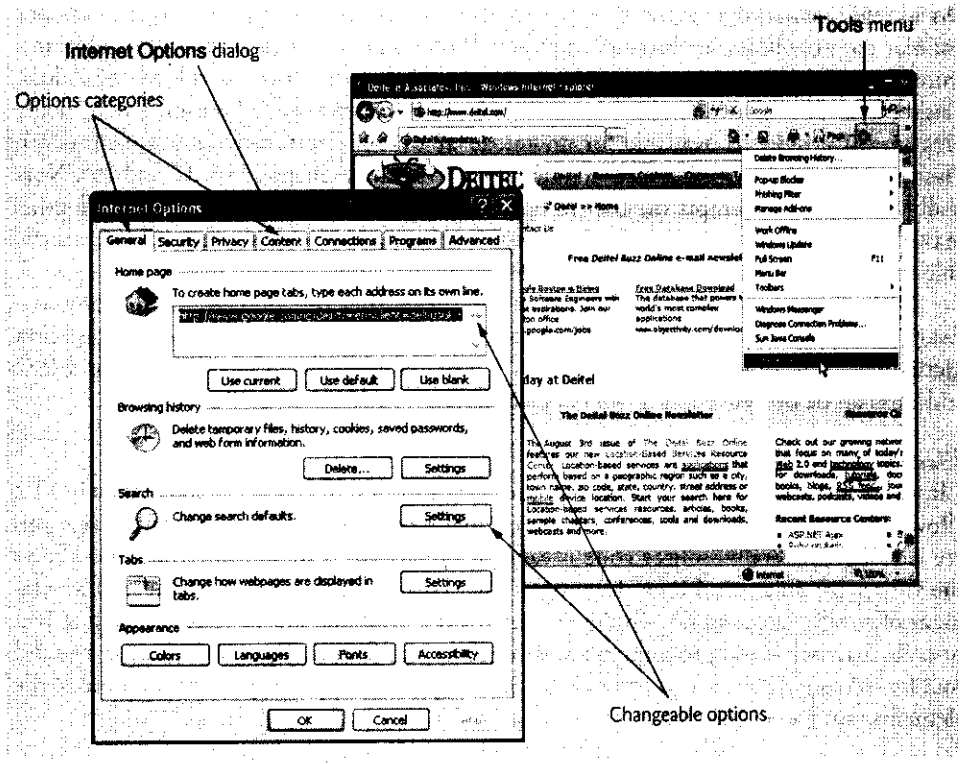

Fig. 2.7 | **Internet Options** in Internet Explorer 7.

History options also may be adjusted in this category. By clicking the **Settings** button in the **Browsing history** section of the **General** tab in IE7, or the **Network** option in the **Advanced** tab of FF2, the amount of disk space to be reserved for the web page **cache** can be set. The cache is an area of temporary storage that a browser designates for saving web pages for rapid future access. When a page is viewed that has been visited recently, IE7 and FF2 check whether they already have some elements on that page (such as images) saved in the cache, to reduce download time. Having a large cache can considerably speed up web browsing, whereas having a small cache saves disk space. Caching can sometimes cause problems, because Internet Explorer and Firefox do not always check to ensure that a cached page is the same as the latest version residing on the web server. Holding down the *Ctrl* key and pressing *F5* in either browser, or pressing *Ctrl*, *Shift* and *R* in FF2, remedies this problem by forcing the browser to retrieve the latest version of the web page from the website. Once the **Internet Options** are set, click **OK** in both browsers.

2.5 Searching the Internet

The Internet provides a wealth of information on virtually any topic. The sheer volume of information on the web can make it difficult for users to find specific information. To help users locate information, many websites provide **search engines** that explore the Internet and maintain searchable records containing information about website content. This section explains how search engines work and discusses two types of search engines.

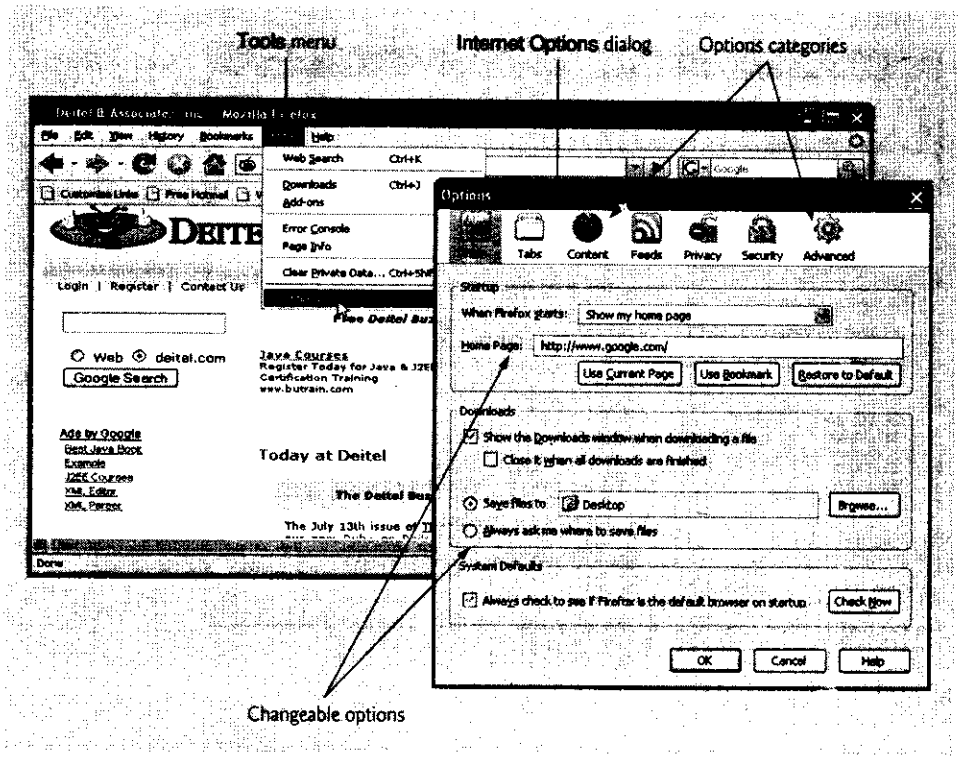

Fig. 2.8 | **Options** in Firefox 2.

Search engines such as Google (`www.google.com`), Yahoo! (`www.yahoo.com`), MSN (`www.msn.com`), AltaVista (`www.altavista.com`) and Ask.com (`www.ask.com`) store information in data repositories called **databases** that facilitate quick information retrieval. When the user enters a word or phrase, the search engine returns a list of hyperlinks to sites that satisfy the search criteria. Each search engine site has different criteria for narrowing searches, such as publishing date, language and relevance. Using multiple search engines may provide the best results in finding the desired content quickly. Sites such as MetaCrawler (`www.metacrawler.com`) use **metasearch engines**, which do not maintain databases. Instead, they send the search criteria to other search engines and aggregate the results. Many web browsers, including IE7 and FF2 (Figs. 2.9–2.10.), have a built-in search box placed in the window that can be used to browse the web. In both browsers, the user can choose which search engine to use by clicking the down-arrow menu (Fig. 2.9–2.10).

Search engines can also be used to help resolve programming errors. There are many websites that contain documentation about specific functions, how to use them correctly and related common errors. Putting a function name or error message into a search engine can often help a programmer discover where a mistake may have been made in the code. Also, websites such as `www.thescripts.com` allow users to post specific programming questions that can be answered by other programmers. Other websites like this one, as well as communities for specific languages, can be found using search engines.

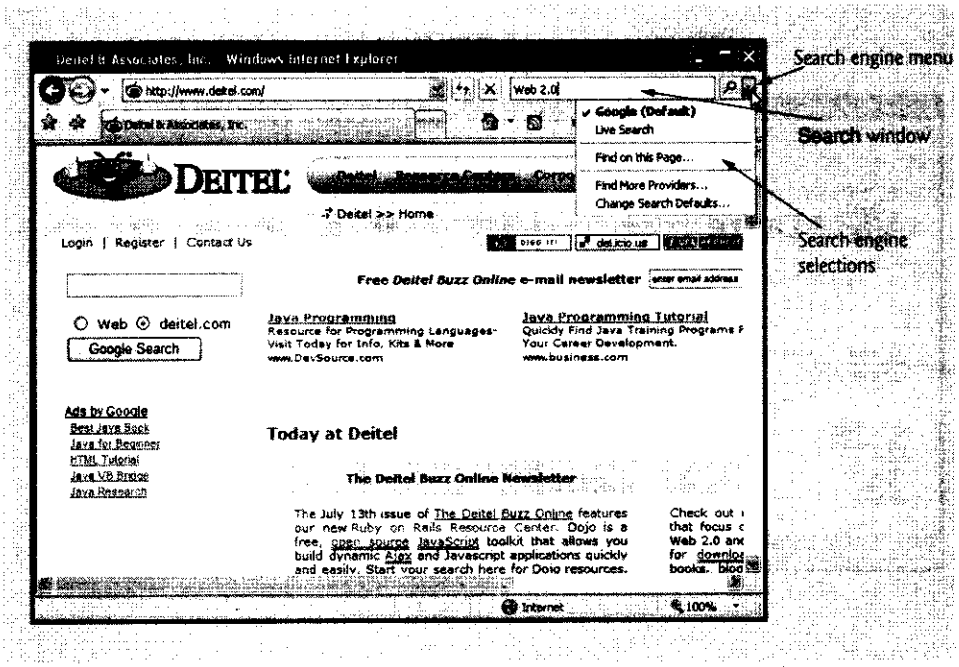

Fig. 2.9 | Searching the Internet with Internet Explorer 7.

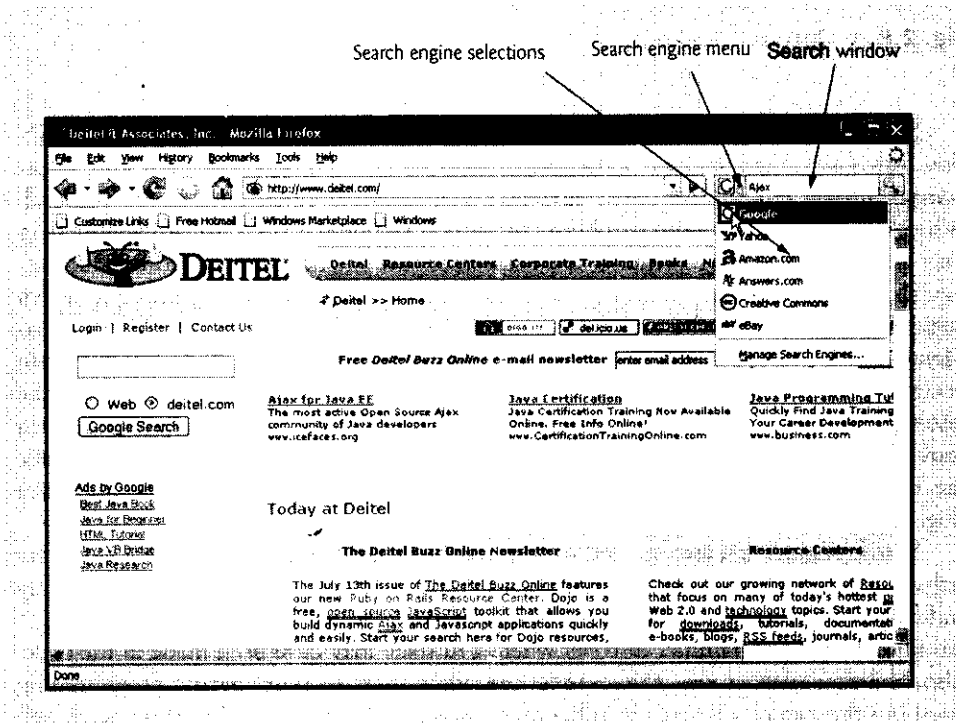

Fig. 2.10 | Searching the Internet with Firefox 2.

2.6 Keeping Track of Your Favorite Sites

As users browse the web, they often visit certain sites repeatedly and may want to record their URL and title. IE7 provides a feature called **favorites** for bookmarking (keeping track of) such sites (Fig. 2.11). Any page's URL can be added to the list of favorites using the **Favorites** menu's **Add to Favorites...** command, or by pressing the yellow star and green plus icon in the upper left corner of the window. A **Favorites** window can also be accessed by clicking the yellow star icon on the toolbar and clicking the **Favorites** option. Favorites can be accessed at any time by selecting them with the mouse from the **Favorites** menu. Favorites can be categorized and grouped into folders in the **Organize Favorites** dialog (displayed when **Organize Favorites...** is selected from the **Favorites** menu). These folders appear as submenus in the **Favorites** menu. The **Organize Favorites** dialog also allows users to rename, delete and move favorites between folders.

FF2 has a similar feature called **bookmarks**, which can be added with the **Bookmark This Page...** option in the **Bookmark** menu and used the same way that **Favorites** are described in this section (Fig. 2.12). Most browsers have their own version of **Favorites** or **Bookmarks**.

2.7 File Transfer Protocol (FTP)

The **File Transfer Protocol** (FTP) is a set of rules by which computers transfer data, especially large files, over the Internet. An FTP site's URL begins with `ftp://` rather than `http://`, and can also be accessed either with the web browser or software that supports FTP. **Filezilla** is a popular, open source FTP client for Windows that functions outside a

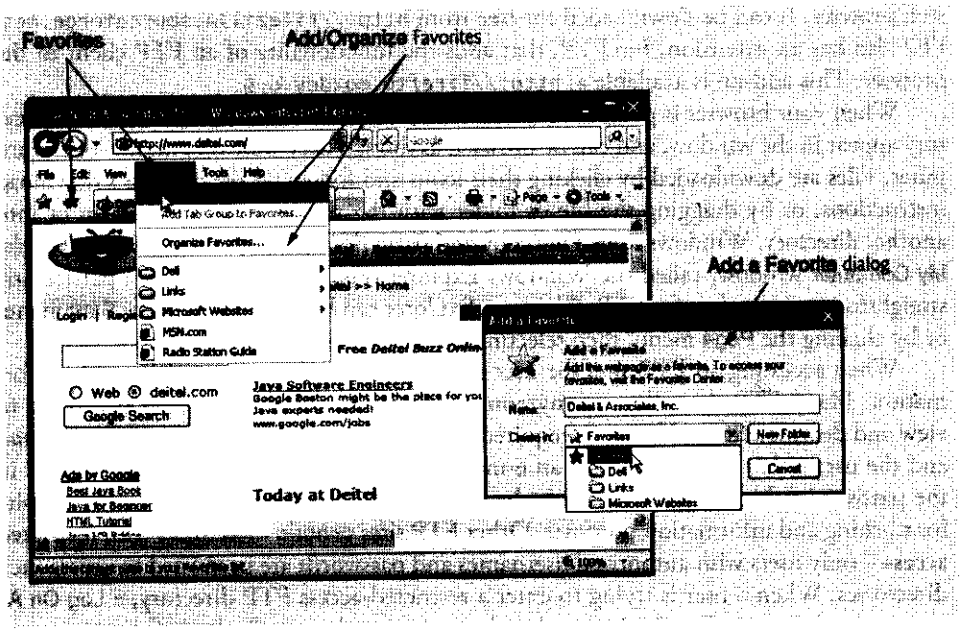

Fig. 2.11 | The **Favorites** menu helps organize frequently visited websites in Internet Explorer 7.

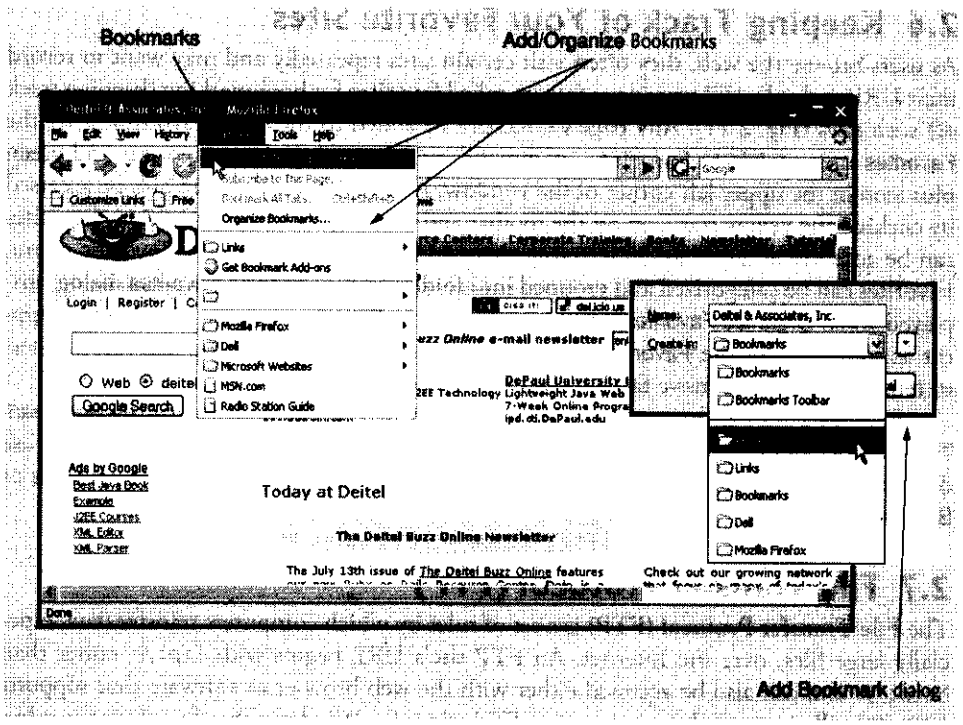

Fig. 2.12 | The **Bookmarks** menu helps organize frequently visited websites in Firefox 2.

web browser. It can be downloaded for free from `http://filezilla.sourceforge.net`. FF2 also has an extension, FireFTP, that adds the functionality of an FTP client to the browser. This add-on is available at `http://fireftp.mozdev.org`.

When your browser is pointed to an FTP site, the contents of the specified site directory appear in the window, and can be browsed as though they were files on the local computer. Files are downloaded by clicking their icons and following the browser's download instructions, or by dragging the file or folder with the mouse onto the desktop or into another directory. Windows users may copy and paste the URL into the address bar of the **My Computer** window, called the Windows Explorer (Fig. 2.13), which has a particularly straightforward interface for FTP. Windows Explorer can be accessed from the **Start** menu, or by clicking the **Page** menu, then selecting **Open FTP Site in Windows Explorer** in IE7.

When accessing an FTP site, the user may or may not be prompted for login information. Many FTP sites allow **anonymous FTP access**, where any user is permitted to view and download files. If login is required, the username is set by default to *anonymous*, and the user either is prompted for an e-mail address or should put an e-mail address in the password field. The browser sends the user's e-mail address and name to the website for tracking and information purposes. Other FTP sites contain directories with **restricted access**—only users with authorized usernames and passwords are permitted to access such directories. When a user is trying to enter a restricted-access FTP directory, a **Log On As** dialog like the one in Fig. 2.13 is displayed, prompting the user for login information.

Transferring a file from the local machine to another location on the Internet is called **uploading** and can be accomplished using the FTP protocol. Files can be transferred from

the local machine (your computer) to the remote machine (server), with an FTP client. The specific instructions for each client are different, but almost all FTP clients allow you to upload, download and perform other file-managing tasks on your data. Understanding FTP is especially important for web developers, since uploading files to a web server is a necessary part of creating a website.

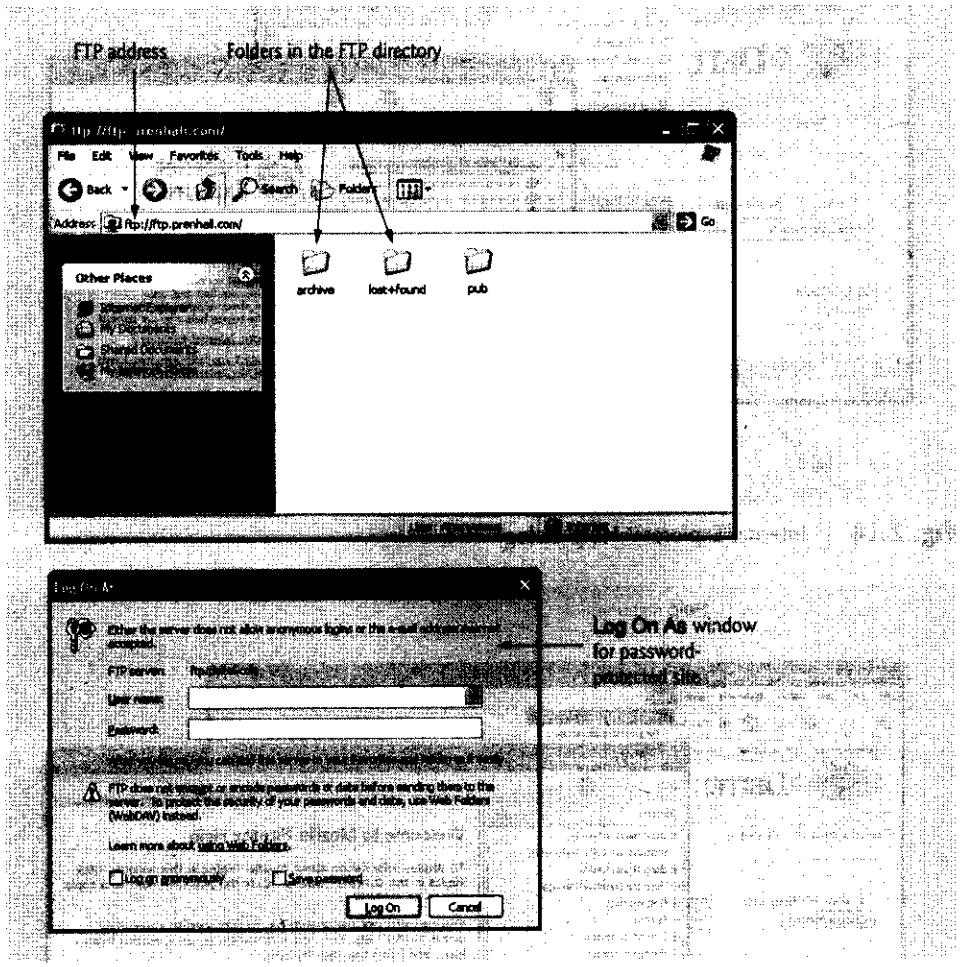

Fig. 2.13 | FTP site access.

2.8 Online Help

Web browsers are complex pieces of software with rich functionality. Although browser designers make every effort to produce user-friendly software, users still need time to familiarize themselves with each web browser and its particular features. Answers to frequently asked questions about using the web browser are included with FF2 and IE7, as well as most other browsers. This information is accessible through the built-in help feature available in the **Help** menu (Figs. 2.14–2.15).

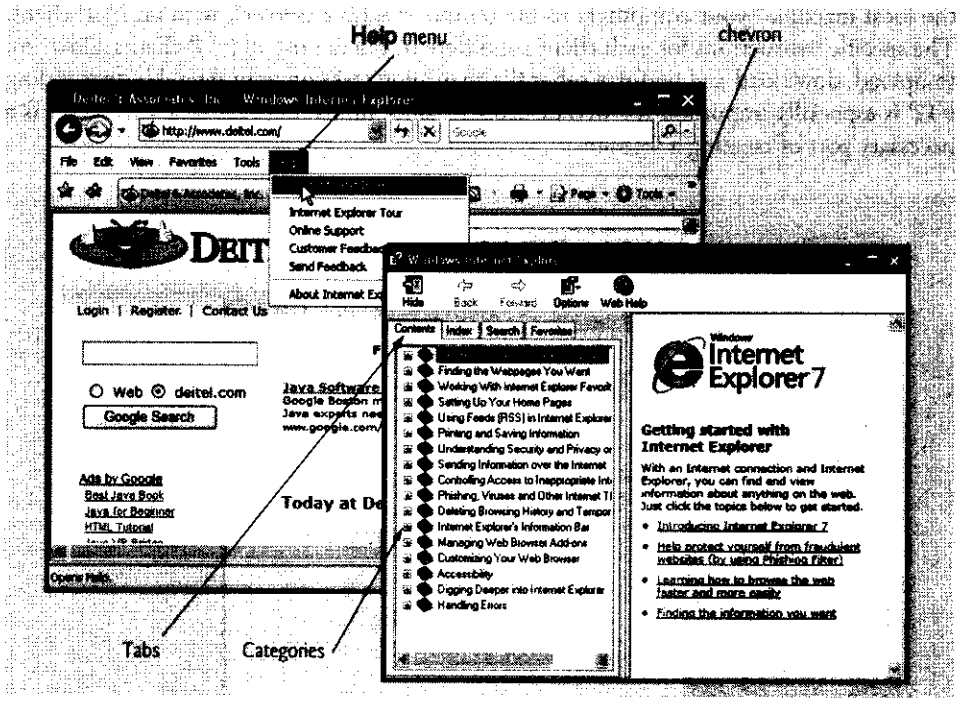

Fig. 2.14 | Internet Explorer 7 **Help** dialog.

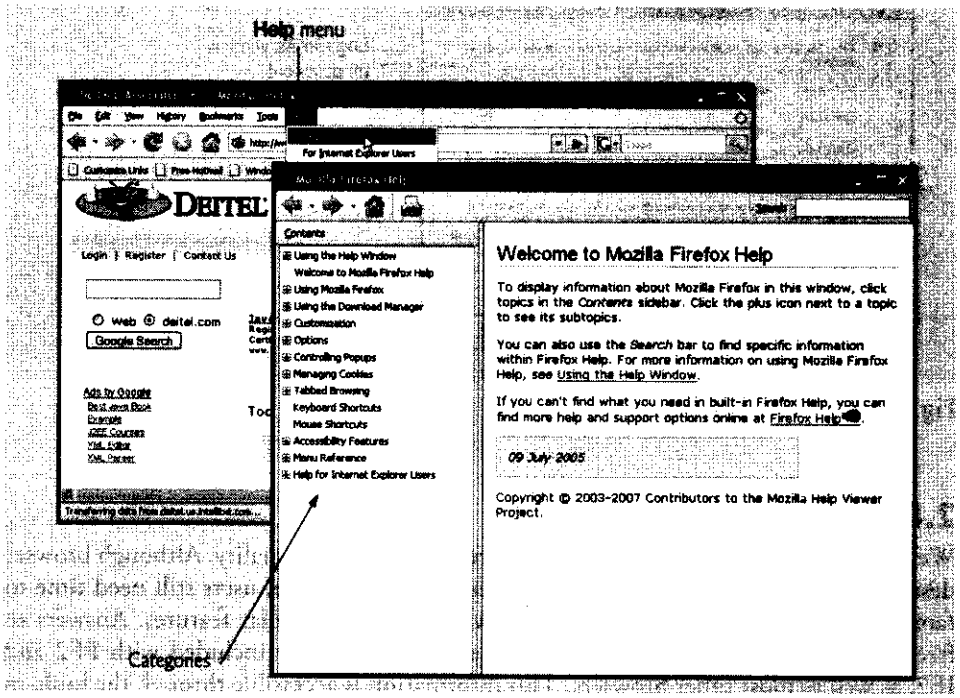

Fig. 2.15 | Firefox 2 **Help** dialog.

A good source for locating help about a specific feature is the **Contents and Index** menu item in IE7 and **Help Contents** in FF2, both accessible through the **Help** menu. IE's help menu can also be accessed by clicking the **Help** option on the toolbar's chevron. When these items are selected, the browser's help dialog is displayed. In IE7, the **Contents** tab organizes the help topics by category, the **Index** tab contains an alphabetical list of **Help** topics and the **Search** tab provides capabilities for searching the help documents. The **Favorites** tab allows users to maintain a list of frequently used help topics. FF2's **Help** window provides a search box and an expandable table of contents for browsing.

2.9 Other Web Browsers

Besides Internet Explorer and Firefox, many other web browsers are available, including *Opera* (`www.opera.com`) and *Safari* (`www.apple.com/safari`). All these browsers differ in functionality, performance and features. Also, they employ different HTML layout engines, which determine how a web page displays in a browser. Firefox 2 uses Gecko as its layout engine, Safari uses a modified version of the KHTML layout engine and Opera and IE7 have their own engines. Gecko and KHTML are both free and open source.

Opera, as well as IE7 and FF2, is a browser designed to be accessible to all users, including those with visual or mobility impairments. Opera software also recently developed a lightweight "Mini" version of the browser that runs effectively on mobile devices. Safari, originally created for Apple's Mac OS, features an elegantly simple interface and impressive speed, especially when executing JavaScript (discussed in Chapters 6–11). Because browsers use different HTML layout engines, they may display the same web page differently. Additionally, some capabilities supported in one browser may not be supported in another. The existence of different browser functionality and features makes cross-browser compatibility difficult to achieve.

2.10 Web Resources

`www.deitel.com/ie7`
`www.deitel.com/firefox`

The Deitel Internet Explorer and Firefox Resource Centers contain links to some of the best resources about these browsers on the web. There you'll find categorized links to Internet Explorer and Firefox downloads and add-ons, keyboard shortcuts, glossaries, code compatibility issues, blogs, forums, podcasts and more. Also check out the range of tutorials on different aspects of these browsers.

Summary

Section 2.1 Introduction to the Internet Explorer 7 and Firefox 2 Web Browsers

- Web browsers are software programs that allow users to access the web's rich multimedia content.
- The two most popular web browsers are Microsoft's *Internet Explorer* and Mozilla's *Firefox*.

Section 2.2 Connecting to the Internet

- A computer alone is not enough to access the Internet. In addition to web browser software, the computer needs specific hardware and a connection to an Internet Service Provider.
- A modem enables a computer to connect to the Internet. A modem converts data to audio tones and transmits the data over phone lines. A network card, also called a network interface card (NIC), allows a computer to connect to the Internet through a network or a high-speed Internet connection, such as a LAN, cable modem or a Digital Subscriber Line (DSL).
- Bandwidth refers to the amount of data that can be transferred through a communications medium in a fixed amount of time. Different ISPs offer different types of high-speed connections, called broadband connections.
- Broadband is a category of high-bandwidth Internet service that is most often provided by cable television and telephone companies to home users.
- DSL is a broadband service that allows computers to be connected at all times to the Internet over existing phone lines, without interfering with voice services.
- Cable modems enable the computer to be connected to the Internet at all times. Cable modems transmit data over the cables that bring television to homes and businesses.
- ISDN provides Internet service over either digital or standard telephone lines. ISDN requires specialized hardware, called a terminal adapter (TA), which is usually obtained from the ISP.
- Fiber optics are a cable alternative that provide greater bandwidth and a better signal, but their popularity is limited by high cost.

Section 2.3 Internet Explorer 7 and Firefox 2 Features

- A URL is the address of a web page. Each web page is associated with a unique URL. URLs usually begin with `http://`, which stands for Hypertext Transfer Protocol (HTTP), the industry standard protocol for transferring web documents over the Internet.
- URLs of websites that handle private information, such as credit card numbers, often begin with `https://`, the abbreviation for Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS), the standard for transferring encrypted data over the Internet.
- Several techniques are available for navigating between different URLs. A user can click the **Address** field and type a web page's URL. The user can then press *Enter* or click **Go** to request the web page located at that URL.
- Another way to navigate the web is via visual elements on web pages called hyperlinks that, when clicked, load a specified web document. Both images and text serve as hyperlinks.
- Hyperlinks can reference other web pages, e-mail addresses and files. If a hyperlink is a `mailto:`*e-mailaddress*, clicking the link loads your default e-mail program and opens a message window addressed to the specified recipient's e-mail address.
- Tabbed browsing allows you to browse multiple pages without cluttering the desktop with many windows.
- When a file is downloaded, it is copied onto the user's computer. Programs, documents, images and sound files are all downloadable files.
- IE7 and FF2 maintain a list of previously visited URLs in chronological order, called the history.
- The **History** window contains heading levels ordered chronologically. Within each time frame headings are alphabetized by site directory name. This window is useful for finding previously visited sites without having to remember the exact URL.

- URLs from the history are displayed in a drop-down list when a user types a URL into the **Address** bar. This feature is called AutoComplete. Any URL from this drop-down list can be selected with the mouse to load the web page at that URL into the browser.
- Web pages can be saved directly to the computer's hard drive for off-line browsing. Select **Save As...** (IE7) or **Save Page As...** (FF2) from the **File** menu at the top of the browser window to save a web page and all its components.
- Individual images from a website can be saved by clicking the image with the right mouse button and selecting **Save Picture As...** (IE7) or **Save Image As...** (FF2) from the displayed context menu.
- Plug-ins are specialized pieces of software that enable the browser to support additional types of content. Normally the browser prompts the user to download a plug-in when a plug-in is needed.
- Extensions enhance the preexisting functionality of the browser.
- Clicking the **View** menu followed by the **Source** option in IE7 or **Page Source** in FF2 allows you to view the source code, or the original code written to create the web page you are viewing.

Section 2.4 Customizing Browser Settings

- IE7 and FF2 have many settings that determine how sites are displayed, how security measures are applied and how outputs are rendered. Many of these can be accessed through the **Tools** menu, then **Internet Options** in IE7, or **Options** in FF2.
- The privacy level for IE7 can be set under the **Privacy** tab of the **Internet Options** dialog. There are six levels of privacy. The most lenient level permits downloading and cookies; the strictest level renders a constant flow of alerts and alarms about browsing security and might prevent certain websites from working correctly.
- Privacy settings for FF2 can be found under the **Privacy** tab of **Options**, which displays options about how data is remembered in the system and when cookies should be accepted.
- Other security options can be found for both browsers under the **Security** tab.
- A personal home page can be specified under the **General** tab of the **Internet Options** dialog in IE7, and **Main** under **Options** in FF2. The home page is the web page that loads when the browser is first opened and appears when the **Home** button at the top of the browser window is clicked.
- History and cache options may be adjusted in the **General** tab of the **Internet Options** dialog by clicking the **Settings...** button in IE7, and the cache can be adjusted in the **Network** option in the **Advanced** tab of FF2. The amount of disk space to be reserved for web-page cache can be set here.

Section 2.5 Searching the Internet

- Search engines explore the Internet and maintain searchable records containing information about websites.
- Metasearch engines do not maintain databases. Instead, they send the search criteria to other search engines and aggregate the results. IE7 and FF2 have built-in search boxes next to the **Address** bar with several different search engines which can be selected by the user.
- Search engines are helpful tools for finding solutions to programming problems.

Section 2.6 Keeping Track of Your Favorite Sites

- As users browse the web, they often visit certain sites repeatedly. Internet Explorer 7 provides a feature called **Favorites** for bookmarking such sites, and Firefox 2 has a similar tool called **Bookmarks**. Sites can be remembered and organized under the **Favorites** menu in IE7 and the **Bookmarks** menu in FF2.

Section 2.7 File Transfer Protocol (FTP)

- FTP (file transfer protocol) is an older protocol for transferring information, especially large files, over the Internet. An FTP site's URL begins with `ftp://` rather than `http://`, and can be accessed through the browser or by any software that supports FTP.
- FTP sites with anonymous access allow any user access. Many FTP sites have restricted access; only users with authorized usernames and passwords are permitted to access such sites.
- Transferring a file from the local machine to another location on the Internet is called uploading and can be accomplished using the FTP protocol.

Section 2.8 Online Help

- The **Help** menu in the browsers allows the user to search or browse for answers to common questions and solutions to problems with the software.

Section 2.9 Other Web Browsers

- Many other browsers are available for download, each with its own set of features and advantages. Two of these browsers are Opera and Safari.

Terminology

Address bar
Adobe Acrobat Reader
Adobe Flash Player
anonymous FTP
applications
AutoComplete
Back button
bandwidth
bookmark
broadband connection
cable modem
cache
context menu
cookie
database
dial-up connection
Digital Subscriber Line (DSL)
download
extensions
Favorites
fiber optics
file transfer
File Transfer Protocol (FTP)
Filezilla
Firefox 2 (FF2)
Flash Content
Forward button
Gecko
Help menu
History
home page
HTML layout engine
hyperlink
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Integrated Services Digital Network (ISDN)
Internet Explorer 7 (IE7)
Internet Options
Internet Service Provider (ISP)
metasearch engine
modem
Mozilla Firefox
network
network card
network interface card (NIC)
off-line browsing
Opera
plug-in
Portable Document Format (PDF)
Privacy
restricted access
Safari
Search
search engine
security level
session
sharing
Silverlight
source code

terminal adaptor (TA)
Uniform Resource Locator (URL)
uploading
web browser
web editor
Windows Internet Explorer

Self-Review Exercises

2.1 Fill in the blanks in each of the following statements:
a) The two most popular web browsers are _____ and _____.
b) A browser is used to view files on the _____.
c) The location of a file on the Internet is called its _____.
d) The element in a web page that, when clicked, causes a new web page to be called up is the _____ when the mouse passes over this element, the mouse pointer changes to a(n) _____ in IE7 and FF2.
e) IE7 and FF2 keep a list of visited URLs called the _____.
f) You can save an image from a web page by right clicking the image and selecting _____ in IE7 or _____ in FF2.
g) The feature of IE7 and FF2 that provides options for completing URLs is called _____.
h) The feature that enables the user to save URLs of frequently visited web pages is _____ in IE7 or _____ in FF2.

2.2 State whether each of the following is *true* or *false*. If the statement is *false*, explain why.
a) Fiber optics cables have a better signal than traditional metal cables, but lower bandwidth.
b) It is not possible to view web pages when not connected to the Internet.
c) Source code debuggers are used to help resolve programming errors.
d) The cache is an area on the hard drive that is used for saving web pages for rapid future access.
e) FTP is a popular Internet mechanism by which files are uploaded and downloaded.
f) You can access any FTP site by logging in with the user name anonymous.

Exercises

2.3 Spell out the following acronyms, and include a brief description of each:
a) HTTP
b) FTP
c) URL
d) DSL
e) PDF
f) ISP

2.4 Go to www.download.com and browse the various extensions and plug-ins available for the browser called Internet Explorer. Choose one to install, and observe what capabilities are provided by the extension. Browse the Internet.

2.5 Go to www.mozilla.org and browse the various extensions and plug-ins available for the browser called Firefox. Choose one to install, and observe what capabilities are provided by the extension. Browse the Internet.

2.6 Download and install the Opera (www.opera.com) and Safari (www.apple.com/safari) browsers. Go to your favorite web sites and try to observe any differences in appearance and functionality.

Dive Into Web 2.0

Network effects from user contributions are the key to market dominance in the Web 2.0 era.

—Tim O'Reilly

Link by link, click by click, search is building possibly the most lasting, ponderous, and significant cultural artifact in the history of humankind: the Database of Intentions.

—John Battelle, *The Search*

Web 2.0 is a massive social experiment...this is an opportunity to build a new kind of international understanding...citizen to citizen, person to person.

—Lev Grossman, *TIME*

One of the powerful things about networking technology like the Internet or the Web or the Semantic Web...is that the things we've just done with them far surpass the imagination of the people who invented them.

—Tim Berners-Lee, interviewed by Peter Moon, *IDG Now*

Objectives

In this chapter you will learn:

- The defining characteristics of Web 2.0.
- Why search is fundamental to Web 2.0.
- How Web 2.0 empowers the individual.
- The importance of collective intelligence and network effects.
- The significance and growth of blogging.
- Social networking, social media and social bookmarking.
- How tagging leads to folksonomies.
- How Web 2.0 technologies enable new applications to be quickly and easily "mashed up" from existing applications.
- What Web 2.0 businesses do.
- Web 2.0 Internet business and monetization models.
- The emerging Semantic Web (the "web of meaning").