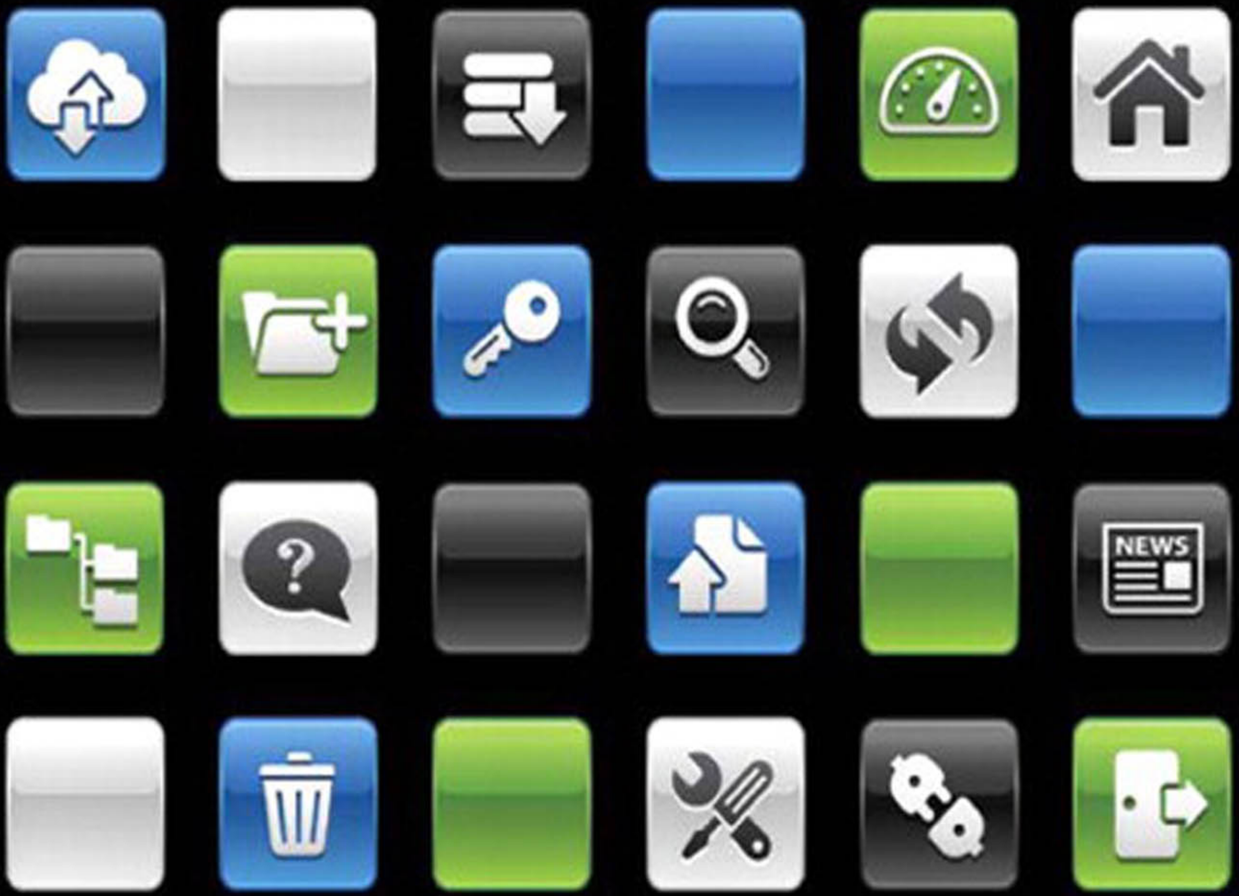
PROGRAMMING THE World Wide Web

EIGHTH EDITION



ROBERT W. SEBESTA

PROGRAMMING THE WORLD WIDE WEB

EIGHTH EDITION

ROBERT W. SEBESTA
University of Colorado at Colorado Springs

PEARSON

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Vice President/Editorial Director: Marcia Horton
Executive Editor: Matt Goldstein
Editorial Assistant: Kelsey Loanes
Senior Managing Editor: Scott Disanno
Program Manager: Kayla Smith-Tarbox
Project Manager: Irwin Zucker
Art Director: Jayne Conte
Cover Designer: Bruce Kenselaar
Cover Art: © Palsur/Shutterstock
Full-Service Project Management: Vasundhara Sawhney/Cenveo® Publisher Services
Composition: Cenveo Publisher Services
Printer/Binder: R.R. Donnelley—Harrisonburg
Cover Printer: R.R. Donnelley—Harrisonburg
Text Font: JansonText

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided "as is" without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services. The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Trademarks

Microsoft® Windows®, and Microsoft Office® are registered trademarks of the microsoft corporation in the U.S.A. And other countries. This book is not sponsored or endorsed by or affiliated with the microsoft corporation.

Many of the designations by manufacturers and seller to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Credits for illustrations appear on page xx.

Library of Congress Cataloging-in-Publication Data
Sebesta, Robert W., author.
Programming the World Wide Web / Robert W. Sebesta, University of Colorado at Colorado Springs. -- Eighth edition.
pages cm
Includes index.
ISBN 978-0-13-377598-3 (alk. paper)
1. Internet programming. 2. World Wide Web. I. Title.

QA76.625.S42 2014
006.7'6--dc 3

2014000161

10 9 8 7 6 5 4 3 2 1

PEARSON

ISBN-10: 0-13-377598-4
ISBN-13: 978-0-13-377598-3

To Aidan

This page intentionally left blank

Preface

It is difficult to overestimate the effect the World Wide Web has had on the day-to-day lives of people, at least those in the developed countries. In just 20 years, we have learned to use the Web for a myriad of disparate tasks, ranging from the mundane task of shopping for airline tickets to the crucial early-morning gathering of business news for a high-stakes day trader.

The speed at which millions of Web sites appeared in the last two decades would seem to indicate that the technologies used to build them were sitting on the shelf, fully developed and ready to use, even before the Web appeared. Also, one might guess that the tens of thousands of people who built those sites were sitting around unemployed, waiting for an opportunity and already possessing the knowledge and abilities required to carry out this mammoth construction task when it appeared. Neither of these was true. The need for new technologies was quickly filled by a large number of entrepreneurs, some at existing companies and some who started new companies. A large part of the programmer need was filled, at least to the extent to which it was filled, by new programmers, some straight from high school. Many, however, were previously employed by other sectors of the software development industry. All of them had to learn to use new languages and technologies.

A visit to a bookstore, either a bricks-and-mortar store or a Web site, will turn up a variety of books on Web technologies aimed at the practicing professional. One difficulty encountered by those teaching courses in Web programming technologies in colleges is the lack of textbooks that are targeted to their needs. Most of the books that discuss Web programming were written for professionals, rather than college students. Such books are written to fulfill the needs of professionals, which are quite different from those of college students. One major difference between an academic book and a professional book lies in the assumptions made by the author about the prior knowledge and experience of the audience. On the one hand, the backgrounds of professionals vary widely, making it difficult to assume much of anything. On the other hand, a book written for junior computer science majors can make some definite assumptions about the background of the reader.

This book is aimed at college students, not necessarily only computer science majors, but anyone who has taken at least two courses in programming. Although students are the primary target, the book is also useful for professional programmers who wish to learn Web programming.

The goal of the book is to provide the reader with a comprehensive introduction to the programming tools and skills required to build and maintain server sites on the Web. A wide variety of technologies are used in the construction of a Web site. There are now many books available for professionals that focus on these technologies. For example, there are dozens of books that specifically address only HTML. The same is true for at least a half-dozen other Web technologies. This book provides descriptions of many of the most widely used Web technologies, as well as an overview of how the Web works.

The first seven editions of the book were used to teach a junior-level Web programming course at the University of Colorado at Colorado Springs. The challenge for students in the course is to learn to use several different programming languages and technologies in one semester. A heavy load of programming exercises is essential to the success of the course. Students in the course build a basic, static Web site, using only HTML as the first assignment. Throughout the remainder of the semester, they add features to their site as the new technologies are introduced in the course. Our students' prior course work in Java and data structures, as well as C and assembly language, is helpful, as is the fact that many of them have learned some HTML on their own before taking the course.

The most important prerequisite to the material of this book is a solid background in programming in some language that supports object-oriented programming. It is helpful to have some knowledge of a second programming language and a bit of UNIX, particularly if a UNIX-based Web server is used for the course. Familiarity with a second language makes learning the new languages easier.

New to the Eighth Edition

- **Chapter 2** Added descriptions of three new type attribute values for the `input` element, `url`, email, and `range` to Section 2.9.2.
- **Chapter 3** Added descriptions of four new selectors, `first-child`, `last-child`, `only-child`, and `empty`, to Section 3.4.5.
- **Chapter 5** Expanded Section 5.9, titled **The `canvas` Element**, from thirteen lines to three and one-half pages, adding three new figures.
- **Chapter 7** Added the new section, 7.2, titled **Uses of XML**, which briefly describes some of the many areas in which XML has been used. Deleted Section 7.4, titled **Document Type Definitions**, in its entirety.
- **Chapter 12** Added Section 12.2.7, titled **Attributes**.
- **Chapter 14** Added a completely new chapter, now Chapter 14, titled **Android Software Development**.

Table of Contents

Chapter 1 lays the groundwork for the rest of the book. A few fundamentals are introduced, including the history and nature of the Internet, the World Wide Web, browsers, servers, URLs, MIME types, and HTTP. Also included in Chapter 1 are brief overviews of the most important topics of the rest of the book.

Chapter 2 provides an introduction to HTML, including images, links, lists, tables, forms, the audio and video elements, the organizational elements, and the time element. Small examples are used to illustrate many of the HTML elements that are discussed in this chapter.

The topic of Chapter 3 is cascading style sheets, which provide the standard way of imposing style on the content specified in HTML tags. Because of the size and complexity of the topic, the chapter does not cover all of the aspects of style sheets. The topics discussed are levels of style sheets, style specification formats, selector formats, property values, and color. Among the properties covered are those for fonts, lists, and margins. Small examples are used to illustrate the subjects that are discussed.

Chapter 4 introduces the core of JavaScript, a powerful language that could be used for a variety of different applications. Our interest, of course, is its use in Web programming. Although JavaScript has become a large and complex language, we use the student's knowledge of programming in other languages to leverage the discussion, thereby providing a useful introduction to the language in a manageably small number of pages. Topics covered are the object model of JavaScript, its control statements, objects, arrays, functions, constructors, and pattern matching.

Chapter 5 discusses some of the features of JavaScript that are related to HTML documents. Included is the use of the basic and DOM 2 event and event-handling model, which can be used in conjunction with some of the elements of HTML documents. The HTML `canvas` element also is described.

One of the interesting applications of JavaScript is building dynamic HTML documents with the Document Object Model (DOM). Chapter 6 provides descriptions of a collection of some of the changes that can be made to documents with the use of JavaScript and the DOM. Included are positioning elements; moving elements; changing the visibility of elements; changing the color, style, and size of text; changing the content of tags; changing the stacking order of overlapped elements; moving elements slowly; and dragging and dropping elements.

Chapter 7 presents an introduction to XML, which provides the means to design topic-specific markup languages that can be shared among users with common interests. Included are the syntax and document structure used by XML, namespaces, XML schemas, and the display of XML documents with both cascading style sheets and XML transformations. Also included is an introduction to Web services and XML processors.

Chapter 8 introduces the Flash authoring environment, which is used to create a wide variety of visual and audio presentations—in particular, those that include animation. A series of examples is used to illustrate the development processes, including drawing figures, creating text, using color, creating motion and shape animations, adding sound tracks to presentations, and designing components that allow the user to control the Flash movie.

Chapter 9 introduces PHP, a server-side scripting language that enjoys wide popularity, especially as a database access language for Web applications. The basics of the language are discussed, as well as the use of cookies and session tracking. The use of PHP as a Web database access language is covered in Chapter 13.

Chapter 10 introduces Ajax, the relatively recent technology that is used to build Web applications with extensive user interactions that are more efficient than those same applications if they do not use Ajax. In addition to a thorough introduction to the concept and implementation of Ajax interactions, the chapter includes discussions of return document forms, Ajax toolkits, and Ajax security. Several examples are used to illustrate approaches to using Ajax.

Java Web software is discussed in Chapter 11. The chapter introduces the mechanisms for building Java servlets and gives several examples of how servlets can be used to present interactive Web documents. The NetBeans framework is introduced and used throughout the chapter. Support for cookies in servlets is presented and illustrated with an example. Then JSP is introduced through a series of examples, including the use of code-behind files. This discussion is followed by an examination of JavaBeans and JavaServer Faces, along with examples to illustrate their use.

Chapter 12 is an introduction to ASP.NET, although it begins with a brief introduction to the .NET Framework and C#. ASP.NET Web controls and some of the events they can raise and how those events can be handled are among the topics discussed in this chapter. ASP.NET AJAX is also discussed. Finally, constructing Web services with ASP.NET is introduced. Visual Studio is introduced and used to develop all ASP.NET examples.

Chapter 13 provides an introduction to database access through the Web. This chapter includes a brief discussion of the nature of relational databases, architectures for database access, the structured query language (SQL), and the free database system MySQL. Then, three approaches to Web access to databases are discussed: using PHP, using Java JDBC, and using ASP.NET. All three are illustrated with complete examples. All of the program examples in the chapter use MySQL.

Chapter 14 introduces the development of Android applications. The basics of view documents, which are written in an XML-based markup language, and activities, which are written in a form of Java, are introduced. Several relatively simple examples are used to illustrate this new approach to building Web applications for mobile devices.

Chapter 15 introduces the Ruby programming language. Included are the scalar types and their operations, control statements, arrays, hashes, methods, classes, code blocks and iterators, and pattern matching. There is, of course, much more to Ruby, but the chapter includes sufficient material to allow the student to use Ruby for building simple programs and Rails applications.

Chapter 16 introduces the Rails framework, designed to make the construction of Web applications relatively quick and easy. Covered are simple document requests, both static and dynamic, and applications that use databases, including the use of scaffolding.

Appendix A introduces Java to those who have experience with C++ and object-oriented programming, but who do not know Java. Such students can learn enough of the language from this appendix to allow them to understand the Java applets, servlets, JSP, and JDBC that appear in this book.

Appendix B is a list of 140 named colors, along with their hexadecimal codings.

Support Materials

Supplements for the book are available at the Pearson Web site www.pearsonhighered.com/sebesta. Support materials available to all readers of this book include

- A set of lecture notes in the form of PowerPoint files. The notes were developed to be the basis for class lectures on the book material.
- Source code for examples

Additional support material, including solutions to selected exercises and figures from the book, are available only to instructors adopting this textbook for classroom use. Contact your school's Pearson Education representative for information on obtaining access to this material, or visit pearsonhighered.com.

Software Availability

Most of the software systems described in this book are available free to students. These systems include browsers that provide interpreters for JavaScript and parsers for XML. Also, PHP, Ruby, and Java language processors, the Rails framework, the Java class libraries to support servlets, the Java JDBC, and the Android Development system, are available and free. ASP.NET is supported by the .NET software available from Microsoft. The Visual Web Developer 2013, a noncommercial version of Visual Studio, is available free from Microsoft. A free 30-day trial version of the Flash development environment is available from Adobe.

Differences between the Seventh Edition and the Eighth Edition

The eighth edition of this book differs from the seventh in the following ways:

Descriptions of the `url`, `email`, and `range` attributes of the `input` element were added to Chapter 2.

Descriptions of four new selectors, `first-child`, `last-child`, `only-child`, and `empty`, were added to Chapter 3.

The description of the `canvas` element was increased from a paragraph to three and one-half pages and three new figures were added to Chapter 5.

A new section was added to Chapter 7, titled **Uses of XML**, which briefly describes some of the many areas in which XML has been used. Section 7.4, titled **Document Type Definitions**, was deleted in its entirety.

A new section, titled **Attributes**, which describes the attributes of C# was added to Chapter 12.

A completely new chapter was added to the book, Chapter 14, titled **Android Software Development**, which introduces the structure of Android applications and the process of developing them. The use of intents to call other activities and data persistence are also discussed.

Throughout the book, numerous small revisions, additions, and deletions were made to improve the correctness and clarity of the material.

Acknowledgments

The quality of this book was significantly improved as a result of the extensive suggestions, corrections, and comments provided by its reviewers. It was reviewed by the following individuals:

Lynn Beighley

R. Blank
CTO, Almer/Blank; Training Director, The Rich Media Institute; Faculty, USC Viterbi School of Engineering

Stephen Brinton
Gordon College

David Brown
Pellissippi State Technical Community College

Barry Burd
Drew University

William Cantor
Pennsylvania State University

Dunren Che
Southern Illinois University Carbondale

Brian Chess
Fortify Software

Randy Connolly
Mount Royal University

Mark DeLuca
Pennsylvania State University

Sanjay Dhamankar
President, OMNIMA Systems, Inc.

Marty Hall

Peter S. Kimble
University of Illinois

Mark Llewellyn
University of Central Florida

Chris Love
ProfessionalASPNET.com

Gabriele Meiselwitz
Towson University

Eugene A. "Mojo" Modjeski
Rose State College

Najib Nadi
Villanova University

Russ Olsen

Jamel Schiller
University of Wisconsin—Green Bay

Stephanie Smullen
University of Tennessee at Chattanooga

Marjan Trutschl
Louisiana State University—Shreveport

J. Reuben Wetherbee
University of Pennsylvania

Christopher C. Whitehead
Columbus State University

Matt Goldstein, Executive Editor; Kelsey Loanes, Editorial Assistant, and Kayla Smith-Tarbox, Program Manager, all deserve my gratitude for their encouragement and help in completing the manuscript.

Brief Contents

Contents

Credits

Figures 2.10, 2.14, and 3.7 Courtesy of Robert W. Sebesta
Figures 2.11 and 2.12 © Total Validator
Figures 3.12, 6.9, and 6.12 © Colin Underhill / Alamy
Figures 4.13, 5.1, 12.5–12.8, 12.21–12.23 © Microsoft Corporation
Figures 4.14 and 5.2 © Mozilla
Figure 4.15 © Google, Inc.
Figure 6.10 © Chris Mattison / Alamy
Figure 6.11 © Charles Polidano / Touch The Skies / Alamy
Figures 8.1–8.5, 8.8, 8.10, 8.13, 8.14, 8.16, 8.19, 8.20 8.23, 8.26–8.33 © Adobe Systems Inc. All rights reserved. Adobe and Flash is/are either [a] registered trademark[s] or a trademark[s] of Adobe Systems Incorporated in the United States and/or other countries.
Figures 11.4–11.9, 11.1711.18 © Oracle and/or its affiliates. All rights reserved.

CHAPTER

1

Fundamentals

The lives of most inhabitants of the industrialized countries, as well as many in the unindustrialized countries, have been changed forever by the advent of the World Wide Web. Although this transformation has had some downsides—for example, easier access to pornography and gambling and the ease with which people with destructive ideas can propagate those ideas to others—on balance, the changes have been enormously positive. Many use the Internet and the World Wide Web daily, communicating with friends, relatives, and business associates through electronic mail and social networking sites, shopping for virtually anything that can be purchased anywhere, and digging up a limitless variety and amount of information, from movie theater show times, to hotel room prices in cities halfway around the world, to the history and characteristics of the culture of some small and obscure society. In recent years, social networking has been used effectively to organize social and political demonstrations, and even revolutions. Constructing the software and data that provide access to all this information requires knowledge of several different technologies, such as markup languages and meta-markup languages, as well as programming skills in a myriad of different programming languages, some specific to the World Wide Web and some designed for general-purpose computing. This book is meant to provide the required background and a basis for acquiring the knowledge and skills necessary to build the World Wide Web sites that provide both the information users want and the advertising that pays for its presentation.

This chapter lays the groundwork for the remainder of the book. It begins with introductions to, and some history of, the Internet and the World Wide Web. Then, it discusses the purposes and some of the characteristics of Web browsers and servers. Next, it describes Uniform Resource Locators (URLs), which specify addresses of resources available on the Web. Following this, it introduces Multipurpose Internet Mail Extensions (MIMEs), which define types and file name extensions for files with different kinds of contents. Next, it discusses the Hypertext Transfer Protocol (HTTP), which provides the communication interface for connections between browsers and Web servers. Finally, the chapter gives brief overviews of some of the tools commonly used by Web programmers, including HTML, XML, JavaScript, Flash, Servlets, JSP, JSF, ASP.NET, PHP, Ruby, Rails, and Ajax. They are discussed in far more detail in the remainder of the book (HTML in Chapters 2 and 3; JavaScript in Chapters 4, 5, and 6; XML in Chapter 7; Flash in Chapter 8; PHP in Chapter 9; Ajax in Chapter 10; Servlets, JSP, and JSF in Chapter 11; Ruby in Chapters 14 and 15; and Rails in Chapter 15).

1.1 A Brief Introduction to the Internet

Virtually every topic discussed in this book is related to the Internet. Therefore, we begin with a quick introduction to the Internet itself.

1.1.1 Origins

In the 1960s, the U.S. Department of Defense (DoD) became interested in developing a new large-scale computer network. The purposes of this network were communications, program sharing, and remote computer access for researchers working on defense-related contracts. One fundamental requirement was that the network be sufficiently robust so that even if some network nodes were lost to sabotage, war, or some more benign cause, the network would continue to function. The DoD's Advanced Research Projects Agency (ARPA)[1] funded the construction of the first such network, which connected about a dozen ARPA-funded research laboratories and universities. The first node of this network was established at UCLA in 1969.

Because it was funded by ARPA, the network was named ARPAnet. Despite the initial intentions, the primary early use of ARPAnet was simple text-based communications through electronic mail. Because ARPAnet was available only

[1] ARPA was renamed Defense Advanced Research Projects Agency (DARPA) in 1972.

to laboratories and universities that conducted ARPA-funded research, the great majority of educational institutions were not connected. As a result, several other networks were developed during the late 1970s and early 1980s, with BITNET and CSNET among them. BITNET, which is an acronym for *B*ecause *I*t's *T*ime *Net*work, began at the City University of New York. It was built initially to provide electronic mail and file transfers. CSNET, which is an acronym for *C*omputer *S*cience *Net*work, connected the University of Delaware, Purdue University, the University of Wisconsin, the RAND Corporation, and Bolt, Beranek, and Newman (a research company in Cambridge, Massachusetts). Its initial purpose was to provide electronic mail. For a variety of reasons, neither BITNET nor CSNET became a widely used national network.

A new national network, NSFnet, was created in 1986. It was sponsored, of course, by the National Science Foundation (NSF). NSFnet initially connected the NSF-funded supercomputer centers that were at five universities. Soon after being established, it became available to other academic institutions and research laboratories. By 1990, NSFnet had replaced ARPAnet for most nonmilitary uses, and a wide variety of organizations had established nodes on the new network—by 1992, NSFnet connected more than one million computers around the world. In 1995, a small part of NSFnet returned to being a research network. The rest became known as the Internet, although this term was used much earlier for both ARPAnet and NSFnet.

1.1.2 What Is the Internet?

The Internet is a huge collection of computers connected in a communications network. These computers are of every imaginable size, configuration, and manufacturer. In fact, some of the devices connected to the Internet—such as plotters and printers—are not computers at all. The innovation that allows all these diverse devices to communicate with each other is a single, low-level protocol named Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP became the standard for computer network connections in 1982. It can be used directly to allow a program on one computer to communicate with a program on another computer via the Internet. In most cases, however, a higher-level protocol runs on top of TCP/IP. Nevertheless, it is TCP/IP that provides the low-level interface that allows most computers (and other devices) connected to the Internet to appear exactly the same.[2]

Rather than connecting every computer on the Internet directly to every other computer on the Internet, normally the individual computers in an organization are connected to each other in a local network. One node on this local network is physically connected to the Internet. So, the Internet is actually a network of networks, rather than a network of computers.

Obviously, all devices connected to the Internet must be uniquely identifiable.

[2] TCP/IP is not the only communication protocol used by the Internet—User Datagram Protocol/Internet Protocol (UDP/IP) is an alternative that is used in some situations.

1.1.3 Internet Protocol Addresses

For people, Internet nodes are identified by names; for computers, they are identified by numeric addresses. This relationship exactly parallels the one between a variable name in a program, which is for people, and the variable's numeric memory address, which is for the machine.

The Internet Protocol (IP) address of a machine connected to the Internet is a unique 32-bit number. IP addresses usually are written (and thought of) as four 8-bit numbers, separated by periods. The four parts are separately used by Internet-routing computers to decide where a message must go next to get to its destination.

Organizations are assigned blocks of IPs, which they in turn assign to their machines that need Internet access—which now include virtually all computers. For example, a small organization may be assigned 256 IP addresses, such as `191.57.126.0` to `191.57.126.255`. Very large organizations, such as the Department of Defense, may be assigned 16 million IP addresses, which include IP addresses with one particular first 8-bit number, such as `12.0.0.0` to `12.255.255.255`.

Although people nearly always type domain names into their browsers, the IP works just as well. For example, the IP for United Airlines (`www.ual.com`) is `209.87.113.93`. So, if a browser is pointed at `http://209.87.113.93`, it will be connected to the United Airlines Web site.

In late 1998, a new IP standard, IPv6, was approved, although it still is not widely used. The most significant change was to expand the address size from 32 bits to 128 bits. This is a change that will soon be essential because the number of remaining unused IP addresses is diminishing rapidly.

1.1.4 Domain Names

Because people have difficulty dealing with and remembering numbers, machines on the Internet also have textual names. These names begin with the name of the host machine, followed by progressively larger enclosing collections of machines, called *domains*. There may be two, three, or more domain names. The first domain name, which appears immediately to the right of the host name, is the domain of which the host is a part. The second domain name gives the domain of which the first domain is a part. The last domain name identifies the type of organization in which the host resides, which is the largest domain in the site's name. For organizations in the United States, `edu` is the extension for educational institutions, `com` specifies a company, `gov` is used for the U.S. government, and `org` is used for many other kinds of organizations. In other countries, the largest domain is often an abbreviation for the country—for example, `se` is used for Sweden, and `kz` is used for Kazakhstan.

Consider this sample address:

```
movies.marxbros.comedy.com
```

Here, `movies` is the host name and `marxbros` is `movies`'s local domain, which is a part of `comedy`'s domain, which is a part of the `com` domain. The host name and all the domain names are together called a *fully qualified domain name*.

Because IP addresses are the addresses used internally by the Internet, the fully qualified domain name of the destination for a message, which is what is given by a browser user, must be converted to an IP address before the message can be transmitted over the Internet to the destination. These conversions are done by software systems called *name servers*, which implement the Domain Name System (DNS). Name servers serve a collection of machines on the Internet and are operated by organizations that are responsible for the part of the Internet to which those machines are connected. All document requests from browsers are routed to the nearest name server. If the name server can convert the fully qualified domain name to an IP address, it does so. If it cannot, the name server sends the fully qualified domain name to another name server for conversion. Like IP addresses, fully qualified domain names must be unique. Figure 1.1 shows how fully qualified domain names requested by a browser are translated into IPs before they are routed to the appropriate Web server.

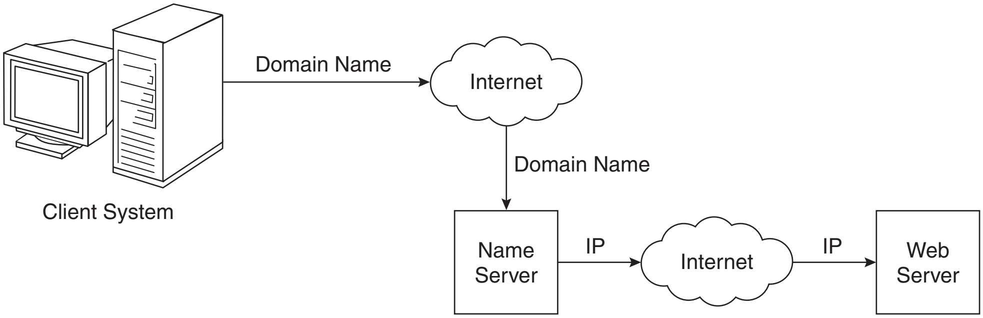


Figure 1.1 Domain name conversion

One way to determine the IP address of a Web site is by using `telnet` on the fully qualified domain name. This approach is illustrated in Section 1.7.1.

By the mid-1980s, a collection of different protocols that run on top of TCP/IP had been developed to support a variety of Internet uses. Among these protocols, the most common were `telnet`, which was developed to allow a user on one computer on the Internet to log onto and use another computer on the Internet; File Transfer Protocol (`ftp`), which was developed to transfer files among computers on the Internet; `Usenet`, which was developed to serve as an electronic bulletin board; and `mailto`, which was developed to allow messages to be sent from the user of one computer on the Internet to other users of other computers on the Internet.

This variety of protocols, each having its own user interface and useful only for the purpose for which it was designed, restricted the growth of the Internet. Users were required to learn all the different interfaces to gain all the advantages of the Internet. Before long, however, a better approach was developed: the World Wide Web.

1.2 The World Wide Web

This section provides a brief introduction to the evolution of the World Wide Web.

1.2.1 Origins

In 1989, a small group of people led by Tim Berners-Lee at Conseil Européen pour la Recherche Nucléaire (CERN) or European Organization for Particle Physics proposed a new protocol for the Internet, as well as a system of document access to use it.[3] The intent of this new system, which the group named the World Wide Web, was to allow scientists around the world to use the Internet to exchange documents describing their work.

The proposed new system was designed to allow a user anywhere on the Internet to search for and retrieve documents from databases on any number of different document-serving computers connected to the Internet. By late 1990, the basic ideas for the new system had been fully developed and implemented on a NeXT computer at CERN. In 1991, the system was ported to other computer platforms and released to the rest of the world.

For the form of its documents, the new system used *hypertext*, which is text with embedded links to text, either in the same document or in another document, to allow nonsequential browsing of textual material. The idea of hypertext had been developed earlier and had appeared in Xerox's NoteCards and Apple's HyperCard in the mid-1980s.

From here on, we will refer to the World Wide Web simply as the *Web*. The units of information on the Web have been referred to by several different names; among them, the most common are *pages*, *documents*, and *resources*. Perhaps the best of these is *documents*, although that seems to imply only text. *Pages* is widely used, but it is misleading in that Web units of information often have more than one of the kind of pages that make up printed media. There is some merit to calling these units *resources*, because that covers the possibility of nontextual information. This book will use *documents* and *pages* more or less interchangeably, but we prefer *documents* in most situations.

Documents are sometimes just text, usually with embedded links to other documents, but they often also include images, sound recordings, or other kinds of media. When a document contains nontextual information, it is called *hypermedia*.

In an abstract sense, the Web is a vast collection of documents, some of which are connected by links. These documents are accessed by Web browsers, introduced in Section 1.3, and are provided by Web servers, introduced in Section 1.4.

[3] Although Berners-Lee's college degree (from Oxford) was in physics, his first stint at CERN was as a consulting software engineer. Berners-Lee was born and raised in London.

1.2.2 Web or Internet?

It is important to understand that the Internet and the Web are not the same thing. The *Internet* is a collection of computers and other devices connected by equipment that allows them to communicate with each other. The *Web* is a collection of software and protocols that has been installed on most, if not all, of the computers on the Internet. Some of these computers run Web servers, which provide documents, but most run Web clients, or browsers, which request documents from servers and display them to users. The Internet was quite useful before the Web was developed, and it is still useful without it. However, most users of the Internet now use it through the Web.

1.3 Web Browsers

When two computers communicate over some network, in many cases one acts as a client and the other as a server. The client initiates the communication, which is often a request for information stored on the server, which then sends that information back to the client. The Web, as well as many other systems, operates in this client-server configuration.

Documents provided by servers on the Web are requested by *browsers*, which are programs running on client machines. They are called browsers because they allow the user to browse the resources available on servers. The first browsers were text based—they were not capable of displaying graphic information, nor did they have a graphical user interface. This limitation effectively constrained the growth of the Web. In early 1993, things changed with the release of Mosaic, the first browser with a graphical user interface. Mosaic was developed at the National Center for Supercomputer Applications (NCSA) at the University of Illinois. Mosaic's interface provided convenient access to the Web for users who were neither scientists nor software developers. The first release of Mosaic ran on UNIX systems using the X Window system. By late 1993, versions of Mosaic for Apple Macintosh and Microsoft Windows systems had been released. Finally, users of the computers connected to the Internet around the world had a powerful way to access anything on the Web anywhere in the world. The result of this power and convenience was explosive growth in Web usage.

A browser is a client on the Web because it initiates the communication with a server, which waits for a request from the client before doing anything. In the simplest case, a browser requests a static document from a server. The server locates the document among its servable documents and sends it to the browser, which displays it for the user. However, more complicated situations are common. For example, the server may provide a document that requests input from the user through the browser. After the user supplies the requested input, it is transmitted from the browser to the server, which may use the input to perform some computation and then return a new document to the browser to inform the user of the results of the computation. Sometimes a browser directly requests the execution of a program stored on the server. The output of the program is then returned to the browser.

Although the Web supports a variety of protocols, the most common one is the HTTP. HTTP provides a standard form of communication between browsers and Web servers. Section 1.7 provides an introduction to HTTP.

The most commonly used browsers are Microsoft Internet Explorer (IE), which runs only on PCs that use one of the Microsoft Windows operating systems,[4] Firefox, and Chrome. The latter two are available in versions for several different computing platforms, including Windows, Mac OS, and Linux. Several other browsers are available, including Opera and Apple's Safari. However, because the great majority of browsers now in use are Chrome, IE, or Firefox, in this book we focus on them.

1.4 Web Servers

Web servers are programs that provide documents to requesting browsers. Servers are slave programs: They act only when requests are made to them by browsers running on other computers on the Internet.

The most commonly used Web servers are Apache, which has been implemented for a variety of computer platforms, and Microsoft's Internet Information Server (IIS), which runs under Windows operating systems. As of October 2013, there were over 150 million active Web hosts in operation,[5] about 65 percent of which were Apache, about 16 percent were IIS, and about 14 percent were `nginx` (pronounced "engine-x"), a product produced in Russia.[6]

1.4.1 Web Server Operation

Although having clients and servers is a natural consequence of information distribution, this configuration offers some additional benefits for the Web. While serving information does not take a great deal of time, displaying information on client screens is time consuming. Because Web servers need not be involved in this display process, they can handle many clients. So, it is both a natural and efficient division of labor to have a small number of servers provide documents to a large number of clients.

Web browsers initiate network communications with servers by sending them URLs (discussed in Section 1.5). A URL can specify one of two different things: the address of a data file stored on the server that is to be sent to the client, or a program stored on the server that the client wants executed and the output of the program returned to the client.

[4] Actually, versions 4 and 5 of IE (IE4 and IE5) were also available for Macintosh computers, and IE4 was available for UNIX systems. However, later versions are available for Windows platforms only.

[5] There were well more than 500 million sites on line.

[6] These statistics are from `www.netcraft.com` and `W3techs.com`.

All the communications between a Web client and a Web server use the standard Web protocol, HTTP, which is discussed in Section 1.7.[7]

When a Web server begins execution, it informs the operating system under which it is running that it is now ready to accept incoming network connections through a specific port on the machine. While in this running state, the server runs as a background process in the operating system environment. A Web client, or browser, opens a network connection to a Web server, sends information requests and possibly data to the server, receives information from the server, and closes the connection. Of course, other machines exist between browsers and servers on the network—specifically, network routers and domain name servers. This section, however, focuses on just one part of Web communication: the server.

Simply put, the primary task of a Web server is to monitor a communications port on its host machine, accept HTTP commands through that port, and perform the operations specified by those commands. All HTTP commands include a URL, which includes the specification of a host server machine. When the URL is received, it is translated into either a file name (in which case the file is returned to the requesting client) or a program name (in which case the program is run and its output is sent to the requesting client). This process sounds pretty simple, but, as is the case in many other simple-sounding processes, there are a large number of complicating details.

All current Web servers have a common ancestry: the first two servers, developed at CERN in Europe and NCSA at the University of Illinois. Currently, the two most common server configurations are Apache running on Linux and Microsoft's IIS running on Windows.

1.4.2 General Server Characteristics

Most of the available servers share common characteristics, regardless of their origin or the platform on which they run. This section provides brief descriptions of some of these characteristics.

The file structure of a Web server has two separate directories. The root of one of these is the *document root*. The file hierarchy that grows from the document root stores the Web documents to which the server has direct access and normally serves to clients. The root of the other directory is the *server root*. This directory, along with its descendant directories, stores the server and its support software.

The files stored directly in the document root are those available to clients through top-level URLs. Typically, clients do not access the document root directly in URLs; rather, the server maps requested URLs to the document root, whose location is not known to clients. For example, suppose that the site name is `www.tunias.com` (not a real site—at least, not yet), which we will assume to be a UNIX-based system. Suppose further that the document root is named `topdocs` and is stored in the `/admin/web` directory, making its address `/admin/web/topdocs`. A request for a file from a client with the URL `http://www.tunias.com/petunias.html` will cause the server to search for the file with the file

[7] Some of these communications use HTTPS, the secure version of HTTP.