

Preface

PURPOSE AND SCOPE

This textbook is designed to provide a careful introduction to key technologies that have been developed as part of the birth and maturation of the World Wide Web. My goal is for students using this book to understand the Web at a fundamental level, much as students who learn assembly language understand computers at such a level. This level of understanding should provide a solid foundation on which to build as students subsequently learn about higher-level web development tools based on the technologies covered here. It should also prepare them well for further study of web technologies, both those that exist today and those that will be developed in the future.

The textbook is designed primarily for use in computer science (CS) courses, but other uses are mentioned later. I assume that the reader has a background roughly equivalent to the first three semesters of an undergraduate CS major. For instance, I expect well-developed skills in at least one programming language, familiarity with Java or the background and ability to learn it quickly from other sources (no Java knowledge is required until the last half of the book), and facility with basic data structures, especially trees.

I have chosen topics so as to treat the subject with reasonable breadth while also allowing for significant depth. With respect to breadth, the textbook focuses on technologies that are unlikely to receive detailed treatment in nonweb CS courses. Conversely, this book covers only lightly a number of topics that, while related to the Web, are not web technologies *per se* and are likely to be covered in other CS courses. For instance, while an appendix describes how to connect a Java-based web application to a database management system (DBMS), the book does not attempt to present SQL or database concepts. Other web-related CS topics that are covered narrowly—that is, primarily as they relate directly to web technologies—include computer networks, software engineering, and security. Finally, because of the emphasis on foundational technologies that are fundamentally web-related, higher-level development tools (such as Macromedia[®] Dreamweaver[®] software) and content presentation tools (such as Macromedia[®] Flash[®] software) are not covered.

Another scope consideration arises from the fact that, especially when it comes to server-side software, several web technologies provide similar capabilities, forming a technology class. For example, the ASP.NET, ColdFusion[®], JSP[™], and PHP technologies all occupy the same server-side software niche, and each is currently in widespread use. Even if time and space allowed all of these technologies to be covered in some depth, I suspect that most students would tire of seeing similar concepts dressed in several different sets of clothes. So I have chosen instead to cover one member of each class in some detail and also to provide a high-level comparison of the example technology with other widely used members of the class. It seems reasonable to expect that a student who understands one technology well will be able to quickly adapt to conceptually related technologies as the need arises in the future.

Along these same lines, for each technology class covered I have chosen to use a Java-based representative as the example for the class. Several factors were significant in this

choice. The Java-based technologies covered in this textbook are available for free download and run on all major operating systems. Also, it seems that most CS students today know Java or a closely related language, so using Java-based software should maximize the time that these students can spend learning web technologies themselves as opposed to learning programming languages. Finally, the significant use of Java-based web technologies in support of many major Web sites would seem to imply that knowledge of these technologies may be directly beneficial to many students when they join real-world development environments.

By limiting its scope as described, my hope is that this book will provide readers with a depth of understanding of foundational web technologies and concepts that will enable them to develop high-quality web applications and avoid many of the common mistakes made by less-knowledgeable web developers. Furthermore, my expectation is that students using this book will be able to quickly learn and adapt to new web technologies as they emerge in the future. I also hope that many of them will be well prepared for further research on core web technologies and to eventually contribute to the development of new technologies. In fact, one of my goals is to provide enough background so that anyone who has read this book should be able to subsequently read and understand (with a reasonable amount of effort) the primary reference sources for the standards and technologies covered. From an instructional point of view, this depth of coverage also allows the instructor to assign some challenging and interesting homework and projects.

While the textbook adopts a CS perspective, many courses taught outside CS departments (for example, in information systems/technologies programs) cover similar topics and may benefit from using this book as either a primary or a reference text. Furthermore, I believe that the book may also be helpful to web development professionals who have not had much formal training in web technologies. In fact, I initially taught myself about the Web on the job at a dot-com, and this book to some extent represents “what I wish I’d known.”

FEATURES

Some of the features of the textbook are:

- Detailed coverage of a wide spectrum of web technologies, including:
 - Hypertext Transport Protocol (HTTP)
 - Extensible HyperText Markup Language (XHTML)
 - Cascading Style Sheets (CSS)
 - JavaScript™ language
 - Document Object Model (DOM)
 - Java servlets
 - Extensible Markup Language (XML)
 - XML namespaces
 - Simple API for XML (SAX)
 - XML Path Language (XPath)
 - Extensible Stylesheet Language Transformations (XSLT)
 - Asynchronous JavaScript and XML (Ajax)

JavaServer Pages™ (JSP) technology, including JavaBeans™ object usage
SOAP

Web Services Definition Language (WSDL)

XML Schema

Java API for XML Remote Procedure Call (JAX-RPC)

- Brief overviews of related technologies, including:
 - Common Gateway Interface (CGI)
 - Active Server Pages (ASP) and ASP.NET
 - PHP
 - ColdFusion technology
- Focus on standards, both formal and de facto.
- Detailed coverage of common features in web servers and browsers, using Apache Tomcat and Mozilla™ software as representative examples.
- Use of student-accessible software, so lab setup may not be necessary:
 - Software discussed and used in examples is available for free download and runs on multiple platforms.
 - Detailed instructions are provided for obtaining, installing, and operating all software.
- Detailed instructions for running server-side software using either the file system or a database management system for persistent storage.
- Ongoing “My Own Blog” case study that illustrates how various technologies can be employed together to build a simple blogging application.
- Extensive use of examples. Virtually every concept covered is illustrated by a concrete example. Examples are often short, providing an uncluttered demonstration of the concept. Larger examples are also given to illustrate interactions and provide context and motivation.
- Three types of end-of-chapter problems:
 - Exercises: short-answer problems that test students’ understanding of content (and, in some cases, their analytical skills).
 - Research and exploration: problems that either direct students to reference materials to learn more about selected topics or ask them to perform various experiments, giving them hands-on experience with topics covered.
 - Projects: generally multipart problems that provide instructors with options, from having students add a small function to code provided by the instructor to writing a fairly extensive application (which may be suitable for assignment to a team of students).
- Comprehensive bibliography of authoritative reference materials, all of which are freely available on the Web. (Bibliographic references appear in square brackets, e.g., [IANA-PORTS].)
- Historical perspective sections, providing context for several key web technologies.

TEXTBOOK PLAN AND COURSE SEQUENCES

The first three chapters are about nonprogramming technologies that are fundamental to understanding communication between web browsers and servers as well as how information

is displayed by browsers. The next two chapters cover software development on the client (browser) side. The final four chapters focus on server-side software development.

The progression is a natural one, but the material is covered in a way that allows significant flexibility in the order of coverage. Chapters 1 and 2 should normally be covered first. The next chapter covered could be either Chapter 3 (some of which is a prerequisite for Chapter 5), Chapter 4 (which is a prerequisite for Chapter 5) or Chapter 6 (which is a prerequisite for the final three chapters). The material on Ajax (Section 7.4) and DOM-based XML processing (Section 7.5) depends on Chapter 5 (and therefore on Chapter 4), but otherwise the material in the final four chapters might be taught before Chapters 3 through 5. I suggest teaching the final four chapters in order, as each chapter depends on the preceding one to some extent.

Each chapter is arranged so that the later sections tend to be those that can be covered briefly or even skipped entirely on a first pass through the material. Similarly, within the longer sections it is generally the case that earlier information is more critical than that found later in the section. My own approach, which seemed to work well when classroom-testing early versions of this textbook, was to allocate a fixed length of time to each chapter (slightly more than one week for each of the first two chapters, one and one-half to two weeks for each of the remaining chapters), start at the beginning of the chapter, and cover as much material as the students could reasonably handle within that time. An alternative would be to allocate as much time as needed for full coverage of selected chapters while skimming material in other chapters. The chapter dependencies mentioned in the preceding paragraph should provide guidance if this approach is adopted; for example, based on these dependences, Chapter 3 might make a good candidate for abbreviated coverage.

Source Files

Source files for most of the examples described in this book are available online at <http://www.prenhall.com/jackson>.

Acknowledgments

This book grew from notes I prepared for a course I taught in the spring of 2002. Several students in that course, especially Brian Blackburn, Matt Hershberger, Alex Mezhinsky, Jon Stanich, and Amy Ulinski, encouraged me to turn those notes into a textbook. If I had known then how much work this would entail, I doubt that I would have started! But I am certain that I would not have begun without their encouraging words. Matt's work with me during a subsequent independent study was also a tremendous help. I also appreciate the comments from a number of students on a preliminary version of this textbook, especially Matt Caporali, Dan Dressler, Bobbie Johnson, and Steve Schwab. Dave Eland, a former professor and colleague without whom I would almost certainly not be a computer scientist, also provided useful feedback on an early manuscript.

I learned a great deal about developing software for the Web while working off and on for Essential Surfing Gear, Inc. from 1996 through 2000. I'm grateful to Merrick Furst for providing both that opportunity and the freedom to research a variety of web technologies, and I thank all of my former colleagues for making esgear such a stimulating and fun place

to work. I also appreciate my “day job” employer, Duquesne University, and my former department chairman, Tom Keagy, for being supportive of my work with esgear, first as a consultant and ultimately full time during a leave of absence from Duquesne. In addition, I greatly appreciate the later sabbatical from Duquesne that allowed me to write the bulk of this book.

Jan Luehe of Sun Microsystems provided the fix given in Appendix A for running a secure server under JWSDP 1.3. Jan also helped to secure the agreement of the JWSDP product team to keep JWSDP 1.3 available for download so that readers of this book should be able to run the examples provided for the foreseeable future.

Several folks at Prentice Hall of course deserve mention. Acquisition editors Kate Hargett and Tracy Dunkelberger along with their assistants Mike Giacobbe and Christianna Lee were extremely helpful in getting the project started and providing reviewer feedback. Carole Snyder guided my work on supplemental materials and also helped keep me in the loop with respect to where the book was in the development process. Marcia Horton was very patient in working out contract details with me, and Barrie Reinhold did a masterful job producing marketing materials. The production staffs, headed by Camille Trentacoste at Prentice Hall and Shelley Creager at Techbooks, were professional and supportive. Thanks especially to copy editor Joseph Fineman for making me sound like a better writer than I am.

My immediate family—Cindy, Rebecca, Peter, Emily, and Benjamin—have all been wonderfully understanding and supportive. Benjamin was especially helpful, providing some of the graphics. Many other extended-family members and friends have also provided encouragement. I hope that those whose names don’t appear here will know that I do appreciate what they’ve done.

Adobe[®] and PostScript[®] are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apache Ant, Apache Axis, Apache Catalina, Apache Coyote, Apache HTTP Server, Apache Jasper, and Apache Tomcat are all trademarks of The Apache Software Foundation.

BEA[®] and WebLogic[®] are registered trademarks of BEA Systems, Inc.

Firefox[™], Mozilla[™], and SeaMonkey[™] are trademarks of the Mozilla Foundation.

GNU[®] is a registered trademark of the Free Software Foundation.

Google[™], Google Groups[™], and Google Maps[™] are trademarks of Google Inc.

Helvetica[™] is a Trademark of Heidelberger Druckmaschinen AG, which may be registered in certain jurisdictions, exclusively licensed through Linotype Library GmbH, a wholly owned subsidiary of Heidelberger Druckmaschinen AG.

IBM[®] and WebSphere[®] are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Linux[®] is a registered trademark of Linus Torvalds.

Mac OS[®] and Macintosh[®] are registered trademarks of Apple Computer, Inc. Safari[™] is a trademark of Apple Computer, Inc.

Macromedia[®], ColdFusion[®], Dreamweaver[®], and Flash[®] are trademarks or registered trademarks of Macromedia, Inc. in the United States and/or other countries.

Microsoft[®], Active X[®], FrontPage[®], JScript[®], Visual Basic[®], Visual InterDev[®], and Windows[®] are either registered trademarks or trademarks of Microsoft Corporation in

xvi Preface

the United States and/or other countries. Screen shots of Microsoft products reprinted by permission from Microsoft Corporation.

Mosaic[™] and NCSA Mosaic[™] are proprietary trademarks of the University of Illinois.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Netscape[®] and Netscape Navigator[®] are registered trademarks of Netscape Communications Corporation in the United States and other countries. LiveWire[™] is also a trademark of Netscape Communications Corporation, which may be registered in other countries.

OMG[™] and OMG Interface Definition Language (IDL)[™] are either registered trademarks or trademarks of Object Management Group, Inc. in the United States and/or other countries.

Opera[™] is a trademark of Opera Software ASA.

Sun[™], J2EE[™], J2SE[™], Java[™], JavaBeans[™], Javadoc[™], JavaScript[™], JavaServer Pages[™], JDBC[™], JDK[™], JSP[™], NetBeans[™], SOAP with Attachments API for Java[™], and Sun Java[™] are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. or other countries.

Times New Roman[®] is a trademark of The Monotype Corporation registered in the U.S. Patent and Trademark Office and may be registered in certain other jurisdictions.

Unicode is a registered trademark of Unicode, Inc.

UNIX[®] is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

W3C[®] is a trademark (registered in numerous countries) and P3P[®] is a registered trademark of the World Wide Web Consortium; marks of W3C are registered and held by its host institutions MIT, ERCIM, and Keio.

WS-I and Web Services-Interoperability Organization are trademarks of the Web Services-Interoperability Organization in the United States and other countries.

All other trademarks are property of their respective owners.