
Introduction

Web applications have come a long way since the release of Netscape Navigator 2.0, the first web browser to have a scripting language embedded in it. That language, JavaScript, allowed developers for the first time to manipulate their web pages directly in the user's browser. From those humble beginnings arose the Document Object Model, a full-fledged specification of standard methods and properties for working with HTML and XML data.

The need for a standard API for working with these types of documents became clear almost immediately after HTML and XML technologies began to catch on with the Internet development community. The World Wide Web Consortium (W3C) realized that the types of operations needed to work with these documents were common enough that they could be specified in a portable way, allowing different document-processing application vendors to offer a standard interface. This, in turn, allowed developers to focus their energies on building their applications rather than figuring out which browser or platform they would need to port their work to.

With this book, you learn to use the DOM to work with HTML and XML data in a variety of settings, including web browsers, server-side applications, and consumer client applications. The DOM is language independent, so you don't need to worry about learning an unfamiliar programming language. The DOM is also available in most of the web browsers in use today, such as Internet Explorer 5 and later and Netscape Navigator 6 and later (and Mozilla 1.0 and later), so finding a place to test your applications should be easy. You also learn the different ways of manipulating documents and their content, how to provide the user with new kinds of user interfaces, and how to help lighten the server's load by moving some processing to the client.

Who Should Read this Book

Every day, XML and HTML are becoming more and more prevalent in an ever-widening variety of applications. Developers who want to quickly create applications that leverage these technologies can use the Document Object Model to save themselves

a lot of time and trouble associated with learning one vendor's proprietary interfaces for working with these kinds of structured documents. In this book, you'll find out how to

- ▶ Work with the common data types and structures needed to create professional DOM applications
- ▶ Discover the structure of a document dynamically and navigate its elements
- ▶ Discover and change any part of a document's content
- ▶ Build DOM applications that work across multiple browsers and other implementations
- ▶ Use the browser-based DOM to create entirely new kinds of user interfaces and navigational elements
- ▶ Offload server-side processing onto the client—for example, sorting data and deriving entirely new data from existing data
- ▶ Embed XML data in web pages and present it to the user at runtime
- ▶ Take advantage of common algorithms and processes to avoid reinventing the wheel
- ▶ Effectively debug DOM applications in browsers and other implementations

How to Use This Book

This book is organized into 12 different chapters in three parts. Part I introduces the DOM, explains some important concepts, and introduces the basic structures that the DOM uses to represent a document. It also introduces the DOM interfaces and reviews their functionalities and purposes. Part II addresses real-world DOM usage, from typical DOM algorithms to the DOM support available in various web browsers and other popular application packages. The subject of debugging DOM-based applications is also covered. Part III examines several practical uses of the DOM and shows examples of how to take advantage of the DOM in your applications, from providing client-side data sorting to performing complex data manipulation directly in the client.

If you aren't already familiar with the DOM or its concepts, then you should read Chapters 1 and 2 for an introduction and high-level overview of the DOM, along with what it specifically does not address and leaves as implementation dependent. Otherwise, you can skip directly to Chapter 3 for a look at how DOM interfaces are used to perform common document operations.

It isn't necessary to have a copy of the W3C DOM specification handy in order to use this book, although you may find it helpful as a reference. You can download a copy in one of many different formats (HTML, PDF, PostScript, etc.) by visiting the W3C on the Web at www.w3.org/DOM.

The different parts of the book are described in detail in the following sections.

Part I: Overview of the W3C Document Object Model

Chapter 1, "Introduction to the DOM," presents a high-level overview of how the DOM is organized and how it came into being. In this chapter you learn the history behind the DOM, the forces shaping its evolution, and the design philosophy that went into making the DOM what it is today.

Chapter 2, "How the DOM Represents Structured Documents," discusses how a DOM implementation represents a structured document to a client application. This chapter introduces basic data types and document structures that all DOM applications need to deal with. It also discusses the DOM document tree and other important theoretical DOM concepts.

Chapter 3 "Navigating and Manipulating Structured Documents," introduces the various DOM Level 1 Core API methods available for inspecting document content, modifying it, and navigating among the document's elements.

Part II: Real-World DOM

Chapter 4, "Algorithms," illustrates some common algorithms encountered when working with the DOM. Read this chapter to learn about basic node traversal, determining the relative positions of nodes, and determining node content.

Chapter 5, "Browsers," examines the current level of DOM support in various browser applications such as Netscape 6, IE 5/6, and Opera 6. This chapter investigates the major differences among the implementations, the proprietary extensions of each, and how to go about creating applications that work across multiple browsers.

Chapter 6, "Applications," explores DOM support in various non-browser applications such as Macromedia Dreamweaver, Apache Xerces, etc. Here we discuss the differences among each, how each evolved, and some common applications.

Chapter 7, "Debugging DOM Applications," illustrates various debugging techniques for working with DOM code in browser and non-browser applications. This chapter examines ways of catching error conditions at runtime, making code easier to debug, and examining document structure at specific points in the application's life.

Part III: Practical Uses of the DOM

Chapter 8, “Providing Automatic Document Navigation,” demonstrates common ways of providing the user with navigation aids in a DOM application. In particular, it shows how to build user interface elements that dynamically populate themselves based on document content and how to respond to user events to help a user maintain their place in the document.

Chapter 9, “Dynamic User Interfaces,” demonstrates how to make rich, dynamic user interfaces using DOM methods. This chapter covers how to dynamically create and check common form elements in response to user events and how to use HTML elements to create entirely new types of user interface controls.

Chapter 10, “Client Processing of Data-Driven Documents,” illustrates some common techniques for document processing on the client side, such as sorting data, reading the contents of a document to derive entirely new data, and performing data calculations on the client.

Chapter 11, “Future Directions for the DOM,” looks at the evolution of the DOM and where it is going. Read this chapter to learn about the changes coming up in DOM Level 3, new technologies such as XPath and XSLT, and techniques that have not yet been addressed in the DOM but might be down the road, such as Transaction-based processing.

The appendix, “DOM Core Level 1 API Reference,” lists the core modules and interfaces of DOM Level 1, their arguments, and their associated exception types.