<<Javascript实战>>

图书基本信息

书名：<<Javascript实战>>

13位ISBN编号：9787802057333

10位ISBN编号：7802057337

出版时间：2009-4

出版时间：开明出版社

作者：David Sawyer McFarland

页数：528

<<Javascript实战>>

前言

Not too long ago, the Web was a pretty boring place. Constructed from plain old HTML, Web pages displayed information and not much else. Folks would click a link and then wait for a new Web page to load——and that was about as interactive as it got.These days, most Web sites are almost as responsive as the programs on a desktop computer, reacting immediately to every mouse-click. And it's all thanks to the subject of the book you're holding——JavaScript.What Is lavaScrJpt?JavaScript is a programming language that lets you supercharge your HTML with animation, interactivity, and dynamic visual effects.JavaScript can make Web pages more useful by supplying immediate feedback. For example, a JavaScript-powered shopping cart page can instantly display a total cost, with tax and shipping, the moment a visitor selects a product to buy. Java-Script can produce an error message immediately after someone attempts to sub-mit a Web form that's missing necessary information.JavaScript's main selling point is its immediacy. It lets Web pages respond instantly to the actions of someone interacting with a page——clicking a link, filling out a form, or merely moving the mouse around the screen. JavaScript doesn't suffer from the frustrating delay associated with server-side programming languages like PHP, which rely on communication between the Web browser and the Web server.Because it doesn't rely on constantly loading and reloading Web pages, JavaScript lets you create Web pages that feel and act more like desktop programs than Web pages.If you've visited Google Maps (http://maps.google. com/), you've seen JavaScript in action. Google Maps lets you view a map of your town (or pretty much anywhere else for that matter), zoom in to get a detailed view of streets and bus stops, or zoom out to get a birds-eye view of how to get across town, the state, or the nation.While there were plenty of map sites before Google, they always required reload-ing multiple Web pages (a usually slow process) to get to the information you wanted. Google Maps, on the other hand, works without page refreshes——it responds immediately to your choices.The programs you create with JavaScript can range from the really simple (like popping up a new browser window with a Web page in it) to full blown Web applications like Google Docs (http://docs.google. com/), which let you create pre-sentations, edit documents, and create spreadsheets using your Web browser with the feel of a program running directly on your computer.A Bit of HistoryInvented by Netscape back in 1995, JavaScript is nearly as old as the Web itself.While JavaScript is well respected today, it has a somewhat checkered past. It used to be considered a hobbyist's programming language, used to add less-than-useful effects messages scrolling across the bottom of a Web browser's status bar like a stock-ticker, or animated butterflies following mouse movements around the page.In the early days of JavaScript, it was easy to find thousands of free JavaScript pro-grams (also called scripts) online, but many of those scripts frequently didn't work in all Web browsers, and at times even crashed browsers.In the early days, JavaScript also suffered from incompatibilities between the two prominent browsers, Netscape Navigator and Internet Explorer. Because Netscape and Microsoft tried to outdo each other's browsers by adding newer and (ostensi-bly) better features, the two browsers often acted in very different ways, making it difficult to create JavaScript programs that worked well in both.Fortunately the worst of those days is nearly gone and contemporary browsers like Firefox, Safari, and Internet Explorer 7 have standardized much of the way they handle JavaScript, making it easier to write JavaScript programs that work for most everyone. (There are still a few incompatibilities among current Web browsers, so you'll need to learn a few tricks for dealing with cross-browser problems. You'll learn how to overcome browser incompatibilities in this book.)In the past several years, JavaScript has undergone a rebirth, fueled by high-profile Web sites like Google, Yahoo, and Flickr, which use JavaScript extensively to create interactive Web applications. There's never been a better time to learn JavaScript. With the wealth of knowledge and the quality of scripts being written, even if you're a beginner you can add sophisticated interaction to your Web site——without becoming a computer scientist.JavaScript Is EverywhereJavaScript isn't just for Web pages, either. It's proven to be such a useful program-ming language that if you learn JavaScript you can create Yahoo Widgets and Apple's Dashboard Widgets, write programs for the iPhone, and tap into the scriptable features of many Adobe programs like Acrobat, Photoshop, Illustrator,and Dreamweaver. In fact, Dreamweaver has always offered clever JavaScript pro-grammers a way to add their own commands to the program.In addition, the programming language for Flash—

—ActionScript——is based on JavaScript, so ifyou learn the basics of JavaScript you'll be well prepared to take on Flash programming projects.JavaScript Doesn't Stand AloneJavaScript isn't any good without the two other pillars of Web design——HTML and CSS. Many programmers talk about the three languages as forming the "layers" of a Web page: HTML provides the structural layer, organizing content like pictures and words in a meaningful way; CSS (Cascading Style Sheets) provides the presen-tational layer, making the content in the HTML look good; and JavaScript adds a behavioral layer, bringing a Web page to life so it interacts with Web visitors.In other words, to master JavaScript you need to have a good understanding of both HTML and CSS.HTML: The Barebones StructureHTML(Hypertext Markup Language)uses simple commands called tags to define the various parts of a Web page. For example, this HTML code creates a simple Web page:It may not be exciting, but this example has all the basic elements a Web page needs. This page begins with a few lines that state what type of document the page is and which standards it conforms to. This document type declaration——doctype for short——also points the Web browser to a de on the Internet that contains defi-nitions for that type of file. HTML actually comes in different versions, and you use a different doctype with each. In this case, the doctype for this page indicates that the page is an HTML document that uses a "strict" version of HTML 4.01.In essence, the doctype tells the Web browser how to display the page. In Internet Explorer, the doctype can even affect how CSS and JavaScript work. With an incorrect or missing doctype, you may end up banging your head against a wall as you discover lots of cross-browser differences with your scripts. In other words,always include a doctype in your HTML.There are four types of HTML commonly used today: HTML 4.01 Transitional,HTML 4.01 Strict, XHTML 1.0 Transitional, and XHTML 1.0 Strict. All four are very much alike, with just slight differences in how tags are written and what tags and attributes are allowed. Most Web page editing programs add an appropriate doctype when you create a new Web page, but ifyou want examples of how each is written, you can find templates for the different types of pages at www.webstandards, org/learn/reference/templates.It doesn't really matter which type of HTML you use. All current Web browsers understand each doctype and can display Web pages using any of the four docu-ment types without problem. Which doctype you use isn't nearly as important as making sure a page validates correctly, as described in the box on page 6.How HTML Tags WorkIn the example on the the previous page, as in the HTML code of any Web page you look at, you'll notice that most commands appear in pairs that surround a block of text or other commands. Sandwiched between brackets, these tags are instructions that tell a Web browser how to display the Web page. Tags are the "markup" part of the Hypertext Markup Language.The starting (opening) tag of each pair tells the browser where the instruction begins, and the ending tag tells it where the instruction ends. Ending or closing tags always include a forward slash (/) after the first bracket symbol ((). For example,the tag (p) marks the start of a paragraph, while (/p) marks its end.For a Web page to work correctly, you must include at least these three tags:oThe (html) tag appears once at the beginning of a Web page (after the doc-type) and again (with an added slash) at the end. This tag tells a Web browser that the information contained in this document is written in HTML, as opposed to some other language. All of the contents of a page, including other tags, appear between the opening and closing (html) tags.Ifyou were to think of a Web page as a tree, the (html) tag would be its trunk.Springing from the trunk are two branches that represent the two main parts of any Web page——the head and the body.oThe head of a Web page, surrounded by (head) tags, contains the title of the page. It may also provide other, invisible information (such as search key-words) that browsers and Web search engines can exploit.In addition, the head can contain information that's used by the Web browser for displaying the Web page and for adding interactivity. You put Cascading Style Sheets, for example, in the head of the document. The head of the docu-ment is also where you often include JavaScript programming and links to Java-Script files.oThe body of a Web page, as set apart by its surrounding (body) tags, contains all the information that appears inside a browser window: headlines, text,pictures, and so on.Within the (body) tag, you commonly find the following tags:oYou tell a Web browser where a paragraph of text begins with a (p) (opening paragraph tag), and where it ends with a (/p) (closing paragraph tag).oThe (strong) tag emphasizes text. If you surround some text with it and its partner tag, (/strong), you get boldface type. The HTML snippet (strong)Warning!(/strong) tells a Web browser to display the word "Warning!" in bold type.oThe (a) tag, or anchor tag, creates a hyperlink in a Web page. When

clicked, a hyperlink——or link——can lead anywhere on the Web. You tell the browser where the link points by putting a Web address inside the (a) tags. For instance, you might type (a href=-"http://www, missingmanuals, com/")Click here!(/a).The browser knows that when your visitor clicks the words "Click here!" it should go to the Missing Manual Web site. The hrefpart of the tag is called an attribute and the URL (the Uniform Resource Locator or Web address) is the value. In this example, http://www, rnissingmanuals, com is the value of the href attribute.CSS: Adding Style to Web PagesHTML used to be the only language you needed to know. You could build pages with colorful text and graphics and make words jump out using different sizes,fonts, and colors. But today, visitors expect more from our Web sites, so you need to turn to a newer, more flexible technology——Cascading Style Sheets (CSS)——to give your pages visual sophistication. CSS is a formatting language that lets you make text look good, build complex page layouts, and generally add style to your site.Think of HTML as merely the language you use to structure a page. It helps iden-tify the stuffyou want the world to know about. Tags like (hl) and (h2) denote headlines and assign them relative importance: a heading 1 is more important than a heading 2. The (p) tag indicates a basic paragraph of information. Other tags provide further structural clues: for example, a (ul) tag identifies a bulleted list (to make a list of recipe ingredients more intelligible, for example).CSS, on the other hand, adds design flair to well-organized HTML content, mak-ing it more beautiful and easier to read. Essentially, a CSS style is just a rule that tells a Web browser how to display a particular element on a page. For example,you can create a CSS rule to make all (hl) tags appear 36 pixels tall, in the Ver-dana font, and the color orange. CSS can do more powerful stuff, too, like add borders, change margins, and even control the exact placement of a page element.When it comes to JavaScript, some of the most valuable changes you make to a page involve CSS. You can use JavaScript to add or remove a CSS style from an HTML tag, or dynamically change CSS properties based on a visitor's input or mouse clicks. For example, you can make a page element appear or disappear sim-ply by changing the CSS display property. To animate an item across the screen,you change the CSS position properties dynamically using JavaScript.Anatomy of a StyleA single style that defines the look of one element is a pretty basic beast. It's essen-tially a rule that tells a Web browser how to format something——turn a headline blue, draw a red border around a photo, or create a 150-pixel-wide sidebar box to hold a list of links. If a style could talk, it would say something like, "Hey Browser,make this look like that." A style is, in fact, made up of two elements: the Web page element that the browser formats (the selector) and the actual formatting instruc-tions (the declaration block). For example, a selector can be a headline, a paragraph of text, a photo, and so on. Declaration blocks can turn that text blue, add a red border around a paragraph, position the photo in the center of the page——the possibilities are endless.Of course, CSS styles can't communicate in nice clear English. They have their own language. For example, to set a standard font color and font size for all paragraphs on a Web page, you'd write the following:This style simply says, "Make the text in all paragraphs——marked with (p) tags——red and 1.5 ems tall." (An em is a unit or measurement that's based on a browser's normal text size.) As Figure I-1 illustrates, even a simple style like this example contains several elements:oSelector. The selector tells a Web browser which element or elements on a page to style——like a headline, paragraph, image, or link. In Figure I-1, the selector (p) refers to the (p) tag, which makes Web browsers format all (p) tags using the formatting directions in this style. With the wide range of selectors that CSS offers and a little creativity, you can gain fine control of your pages' formatting.(Selectors are so important, you'll find a detailed discussion of them sarting on page 172.)oDeclaration Block. The code following the selector includes all the formatting options you want to apply to the selector. The block begins with an opening brace ({) and ends with a closing brace (}).oDeclaration. Between the opening and closing braces of a declaration, you add one or more declarations, or formatting instructions. Every declaration has two parts, a property and a value, and ends with a semicolon.oProperty. CSS offers a wide range of formatting options, called properties. A property is a word——-or a few hyphenated words——indicating a certain style effect. Most properties have straightforward names like font-size, margin-top,and background-color. For example, the background-color property sets——you guessed it——a background color.oValue. Finally, you get to express your creative genius by assigning a value to a CSS property——by making a background blue, red, purple, or chartreuse, for example. Different CSS properties require specific types of values

——a color (like red, or #FF0000), a length (like 18px, 2in, or 5em), a URL (like images/background.gif), or a specific keyword (like top, center, or bottom).You don't need to write a style on a single line as pictured in Figure I-1. Many styles have multiple formatting properties, so you can make them easier to read by breaking them up into multiple lines. For example, you may want to put the selectorand opening brace on the first line, each declaration on its own line, and the clos-hag brace by itself on the last line, like so:it's also helpful to indent properties, with either a tab or a couple of spaces, to visi-bly separate the selector from the declarations, making it easy to tellwhich is which. And finally, putting one space between the colon and the property value is optional, but adds to the readability of the style. In fact you can put as much white space between the two as you want. For example color:red, color: red, and color: red all work.Software for JavaScript ProgrammingTo create Web pages made up of HTML, CSS, and JavaScript, you need nothing more than a basic text editor like Notepad (Windows) or Text Edit (Mac). But after typing a few hundred lines of JavaScript code, you may want to try a pro-gram better suited to working with Web pages. This section lists some common programs, both free and those you can buy..Free ProgramsThere are plenty of free programs out there for editing Web pages and style sheets.If you're still using Notepad or Text Edit, give one of these a try. Here's a short list to get you started:oNotepad++ (Windows, http://notepad-plus.sourceforge, net) is a coder's friend. It highlights the syntax of JavaScript and HTML code, and lets you save macros and assign keyboard shortcuts to them so you can automate the process of inserting the code snippets you use most.oHTML-Kit (Windows, www. chami, com/html-kit) is a powerful HTML/XHTML editor that includes lots of useful features, like the ability to preview a Web page directly in the program (so you don't have to switch back and forth between browser and editor), shortcuts for adding HTML tags, and a lot more.oCoffeeCup Free HTML Editor (Windows, www. coffeecup.com/free-editor) is the free version of the commercial ($49) CoffeeCup HTML editor.TextWrangler (Mac, www. barebones, com/products/textwrangler) is free software that's actually a paired down version of BBEdit, the sophisticated, well-known text editor for the Mac. TextWrangler doesn't have all of BBEdit's built-in HTML-tools, but it does include syntax-coloring (highlighting tags and proper-ties in different colors so it's easy to scan a page and identify its parts), FTP support (so you can upload files to a Web server), and more.Commercial SoftwareCommercial Web site development programs range from inexpensive text editors to complete Web site construction tools with all the bells and whistles:oEditPlus (Windows, www. editplus, com) is an inexpensive ($30) text editor that includes syntax-coloring, FTP, auto-completion, and other wrist-saving features.oCoffeCup (Windows, www. coffeecup, com) is a combination text and visual edi-tor ($30). You can either write straight HTML code or use a visual interface to build your pages.oskEdit (Mac, www. skti. org) is a cheap ($25) Web page editor, complete with FTP/SFTP support, code hints, and other useful features.otextMate (Mac, http://macromates, com) is the new darling of Mac program-mers. This text editor ($63) includes many timesaving features for JavaScript programmers like "auto-paired characters," which automatically plops in the second character of a pair of punctuation marks (for example, the program automatically inserts a closing parenthesis after you type an opening parenthesis).oBBEdit (Mac, www. barebones, com/products/bbedit). This much-loved Mac text editor ($125) has plenty of tools for working with HTML, XHTML, CSS, Java-Script, and more. Includes many useful Web building tools and shortcuts.oDreamweaver (Mac and Windows, wvew. macromedia, com/software/drearnweaver) is a visual Web page editor ($399.) It lets you see how your page looks in a Web browser. The program also includes a powerful text-editor for writing JavaScript programs and excellent CSS creation and management tools.Check out Dreamweaver: The Missing Manual for the full skinny on how to use this powerful program.oExpression Web Designer (Windows, www. microsoft, com) is Microsoft's new entry in the Web design field ($299). It replaces FrontPage and includes many professional Web design tools, including excellent CSS features.About This BookUnlike a piece of software such as Microsoft Word or Dreamweaver, JavaScript isn't a single product developed by a single company. There's no support depart-ment at JavaScript headquarters writing an easy-to-read manual for the average Web developer. While you'll find plenty of information on sites like Mozilla.org (see, for example, http://developer, mozilla, org/en/docs/Core_JavaScript_1.5_Reference) or Ecmascript.org (www. ecmascript, org/docs.php), there's no definitive source of information on the JavaScript programming

language.Because there's no manual for JavaScript, people just learning JavaScript often don't know where to begin. And the finer points regarding JavaScript can trip up even seasoned Web pros. The purpose of this book, then, is to serve as the manual that should have come with JavaScript. In this book's pages, you'll find step-by-step instructions for using JavaScript to create highly interactive Web pages.JavaScript: The Missing Manual is designed to accommodate readers who have some experience building Web pages. You'll need to feel comfortable with HTML and CSS to get the most from this book, since JavaScript often works closely with HTML and CSS to achieve its magic. The primary discussions are written for advanced-beginner or intermediate computer users. But if you're new to building Web pages, special boxes called Up to Speed provide the introductory information you need to understand the topic at hand. If you're an advanced Web page jockey,on the other hand, keep your eye out for similar shaded boxes called Power Users'Clinic. They offer more technical tips, tricks, and shortcuts for the experienced computer fan.This Book's Approach to JavaScriptJavaScript is a real programming language: It doesn't work like HTML or CSS, and it has its own set of (often complicated) rules. It's not always easy for Web design-ers to switch gears and start thinking like computer programmers, and there's no one book that can teach you everything there is to know about JavaScript.The goal of JavaScript: The Missing Manual isn't to turn you into the next great programmer. This book is meant to familiarize Web designers with the ins and outs of JavaScript and then move on to advanced tools for adding really useful interactivity to a Web site as quickly and easily as possible.In this book, you'll learn the basics of JavaScript and programming; but just the basics won't make for very exciting Web pages. It's not possible in 400 pages to teach you everything about JavaScript that you need to know to build sophisticated,interactive Web pages. Instead, this book shows you how to use professional (and free) JavaScript code that will liberate you from all of the minute, time-consuming details of creating JavaScript programs that run well across different browsers.You'll learn the basics of JavaScript, and then jump immediately to advanced Web page interactivity with a little help——OK, a lot of help——from some very sophisti-cated but easy-to-use JavaScript helper programs. Think of it this way: You couldbuild a house by cutting down and milling your own lumber, constructing your own windows, doors and doorframes, manufacturing your own tile, and so on.That "do it yourself" approach is common to a lot of JavaScript books. But who has that kind of time? This book's approach is more like building a house by tak-ing advantage of already built pieces and putting them together using basic skills.The end result will be a beautiful and functional house built in a fraction of the time it would take you to learn every step of the process.And even if you want to learn every step of the process, this book is the best place to start. It points out other useful and more advanced JavaScript books so you can continue your programming education after you're done with this book (but only ifyou want to!).About the OutlineJavaScript: The Missing Manual is divided into four parts, each containing several chapters:oPart 1, Getting Started with JavaScript starts at the very beginning. You'll learn the basic building blocks of JavaScript as well as get some helpful tips on com-puter programming in general. This section teaches you how to add a script to a Web page, store and manipulate information, and add smarts to a program so it can respond to different situations. You'll also learn how to communicate with the browser window, store and read cookies, respond to various events like mouse clicks and form submissions, and modify the HTML of a Web page.oPart 2, Building Web Page Features, provides many real-world examples of JavaScript in action. You'll learn how to create pop-up navigation bars, enhance HTML tables, and build an interactive photo gallery. You'll make your Web forms more usable by adding form validation (so visitors can't submit forms missing information), add a calendar widget to make selecting dates easy, and change form options based on selections a Web visitor makes. Finally, you'll create interesting user interfaces with tabbed panels, accordion panels and pop-up dialog boxes that look great and function flawlessly.oPart 3, Ajax: Communicating with the Web Server, covers the technology that single-handedly made JavaScript one of the most glamorous Web languages to learn. In this section, you'll learn how to use JavaScript to communicate with a Web server so your pages can receive information and update themselves based on information provided by a Web server——without having to load a new Webpage.oPart 4, Troubleshooting, Tips, and Tricks, helps you with those times when nothing seems to be working: your perfectly crafted JavaScript program just doesn't seem to do what you want (or worse, it doesn't work at all!). You'll learn the most common errors new programmers make as well as

techniques for discovering and fixing bugs in your programs. In addition, you'll learn a few tips to make your programming more efficient and your scripts run faster.At the end of the book, an appendix provides a detailed list of references to aid you in your further exploration of the JavaScript programming language.Living ExamplesThis book is designed to get your work onto the Web faster and more profession-ally; it's only natural, then, that half the value of this book also lies on the Web.As you read the book's chapters, you'll encounter a number of living examples——step-by-step tutorials that you can build yourself, using raw materials (like graph-ics and half-completed Web pages) that you can download from either www.sawmac, com/javascript/ or from this book's "Missing CD" page at www.missingrnanuals, com/cds. You might not gain very much from simply reading these step-by-step lessons while relaxing in your porch hammock. But if you take the time to work through them at the computer, you'll discover that these tutorials give you unprecedented insight into the way professional designers build Web pagesYou'll also find, in this book's lessons, the URLs of the finished pages, so that you can compare your work with the final result. In other words, you won't just see pictures of JavaScript code in the pages of the book; you'll find the actual, working Web pages on the Internet.About MissJngManuals.comAt www. missingmanuals, com, you'll find articles, tips, and updates to JavaScript:The Missing Manual. In fact, we invite and encourage you to submit such correc-tions and updates yourself. In an effort to keep the book as up to date and accu-rate as possible, each time we print more copies of this book, we'll make any confirmed corrections you've suggested. We'll also note such changes on the Web site, so that you can mark important corrections into your own copy of the book, if you like. (Go to http://missingmanuals, com/feedback, choose the book's name from the pop-up menu, and then click Go to see the changes.)Also on our Feedback page, you can get expert answers to questions that come to you while reading this book, write a book review, and find groups for folks who share your interest in JavaScript.While you're there, sign up for our free monthly email newsletter. Click the "Sign Up for Our Newsletter" link in the left-hand column. You'll find out what's hap-pening in Missing Manual land, meet the authors and editors, get bonus video and book excerpts, and so on.The Very BasicsTo use this book, and indeed to use a computer, you need to know a few basics.This book assumes that you're familiar with a few terms and concepts:oClicking. This book gives you three kinds of instructions that require you to use your computer's mouse or trackpad. To click means to point the arrow cursor at something on the screen and then——without moving the cursor at all——to press and release the clicker button on the mouse (or laptop trackpad). To right-click means to do the same thing with the right mouse button. To double-click, of course, means to click twice in rapid succession, again without moving the cur-sor at all. And to drag means to move the cursor while pressing the button.When you're told to 8g-click something on the Mac, or Ctrl-click something on a PC, you click while pressing the or Ctrl key (both of which are near the Space bar).oMenus. The menus are the words at the top of your screen or window: File, Edit,and so on. Click one to make a list of commands appear, as though they're writ-ten on a window shade you've just pulled down.oKeyboard shortcuts. If you're typing along in a burst of creative energy, it's sometimes disruptive to have to take your hand off the keyboard, grab the mouse, and then use a menu (for example, to use the Bold command). That's why many experienced computer mavens prefer to trigger menu commands by pressing certain combinations on the keyboard. For example, in the Firefox Web browser, you can press Ctrl-+ (Windows) or -+ (Mac) to make text on a Web page get larger (and more readable). When you read an instruction like "press -B," start by pressing the key; while it's down, type the letter B, and then release both keys.oOperating-system basics. This book assumes that you know how to open a pro-gram, surf the Web, and download files. You should know how to use the Start menu (Windows) and the Dock or J menu (Macintosh), as well as the Control Panel (Windows), or System Preferences (Mac OS X).If you've mastered this much information, you have all the technical background you need to enjoy JavaScript: The Missing Manual.About→These→ArrowsThroughout this book, and throughout the Missing Manual series, you'll find sen-tences like this one: "Open the System→Library→Fonts folder." That's short-hand for a much longer instruction that directs you to open three nested folders in sequence, like this: "On your hard drive, you'll find a folder called System. Open that. Inside the System folder window is a folder called Library; double-click it to open it. Inside that folder is yet another one called Fonts. Double-click to open it,too."Similarly, this kind of arrow

shorthand helps to simplify the business of choosing commands in menus, as shown in Figure I-2.Safari Books OnlineWhen you see a Safari Books Online icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.Safari offers a solution that's better than e-Books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you nee the most accurate, current infor-mation. Try it free at http://safari, oreilly, com

<<Javascript实战>>

内容概要

《JavaScript实战》。
一个现代网站需要传递的内容不止是文字信息。
网站的访客期望智能表格、导航标签、弹出式帮助，还有互动的图片浏览。
《JavaScript实战》带来你正好需要的知识，用有风格而且优雅的方式来发布这些组件 —— 即使你不是一个编程高手。

作者简介

作者：(美国)David Sawyer McfarlandDavid Pogue是《纽约时报》的技术专栏作家、畅销书作者和The Missing Manual系列图书的创始人。

<<Javascript实战>>

书籍目录

章节摘录

插图：It may not be exciting, but this example has all the basic elements a Web page needs. This page.begins with a few lines that state what type of document the page is and which standards it conforms to. This document type declaration——-doctype for short——also points the Web browser to a file on the Internet that contains definitions for that type of file. HTML actually comes in different versions, and you use a different doctype with each. In this case, the doctype for this page indicates that the page is an HTML document that uses a "strict" version of HTML 4.01. In essence, the doetype tells the Web browser how to display the page. In Internet Explorer, the doctype can even affect how CSS and JavaScript work. With an incorrect or missing doctype, you may end up banging your head against a wall as you discover lots of cross-browser differences with your scripts. In other words, always include a doctype in your HTMLThere are four types of HTML commonly used today: HTML 4.01 Transitional, HTML 4.01 Strict, XHTML 1.0 Transitional, and XHTML 1.0 Strict. All four are very much alike, with just slight differences in how tags are written and what tags and attributes are allowed. Most Web page editing programs add an appropriate doctype when you create a new Web page, but if you want examples of how each is written, you can find templates for the different types of pages at www.webstandards.org/learn/reference/templates.It doesn't really matter which type of HTML you use. All current Web browsers understand each doctype and can display Web pages using any of the four document types without problem. Which doctype you use isn't nearly as important as making sure a page validates correctly, as described in the box on page 6.

<<Javascript实战>>

编辑推荐

《JavaScript实战(影印版)》讲述了：您需要了解的重要内容开始就构建结构良好并且能和你的HTML和CSS网页协同运行的JavaScript程序。
定制你的代码——《JavaScript实战(影印版)》循序渐进的指导让你增加网页正好要调用的功能。
通过查看《JavaScript实战(影印版)》的实时在线示例开发你自己的网站解决方案。
避免常见错误——找出什么是应该避免的用法。
用预先写好的大量jQuery函数库代码来节省时间和精力。
使用基本的Ajax编程整合来自一台网络服务器或者Google地图的实时数据。
为什么创建Missing Manual系列丛书？
当内容生动、清楚和有趣时，人们能获得最佳的学习效果。
不幸的是，绝大多数计算机图书读起来都像乏味的购物目录。
这就是我为什么开发The Missing Manual系列图书。
这一系列图书有趣，也不怕提到无用或者运行不正确的功能，而且——哦，顺便提一下——是由有实践经验的作者写成的。
在书的每一页上我们都回答这个直接的问题：“ 这个功能有什么用处？
”DavidAPogue是《纽约时报》的技术专栏作家、畅销书作者和The MissingAManual系列图书的创始人。

版权说明