

## *about this book*

---

Like JSTL, this book is aimed at both programmers and nonprogrammers. Parts 1, 2, and 3 are accessible to page authors who start out with nothing more than HTML. Part 4 is intended for Java programmers—and ambitious page authors who want to learn more about how JSTL works behind the scenes.

In part 1, we look broadly at the Web and at two technologies that are important foundations to JSTL: JavaServer Pages (JSP) and the Extensible Markup Language (XML).

In part 2, we delve into JSTL's depths. Chapters 3, 4, and 5 lay the necessary groundwork by discussing JSTL's expression language, conditions, and loops. Chapters 6 through 10 discuss the more exciting features of JSTL: database access, XML manipulation, text importing and formatting, and so forth.

In part 3 (chapters 11, 12, and 13), we look at progressively more complete and integrated examples of JSTL in action. We start with common, stand-alone tasks and move to an example of organizing an entire site—a web portal—using JSTL.

Finally, part 4 discusses how to configure JSTL, integrate Java code, and even write custom tags using JSTL's API.

The appendices contain reference material. Appendix A is a brief summary of all of JSTL's tags. Appendix B lists JSTL's API and goes into detail about some of its advanced features. Appendix C describes the basics of SQL to help you follow some of the book's examples, and appendix D lists online and printed references.

## How to approach the book

---

If you're a web-page author who knows HTML, you'll probably want to start at the beginning. Chapters 1 and 2 will be particularly useful to you, and you can read the rest of the book in order, stopping somewhere around chapter 14 if the material becomes less interesting to you. If you already know JavaScript, pay special attention to chapter 3, because you'll need to master the details of JSTL's expression language. JavaScript won't help you produce dynamic server-side logic in this environment. If you don't know SQL, appendix C will help you follow the book's database examples.

If you're an experienced Java programmer looking to master JSTL in order to use or teach it, you can probably skip part 1. You might want to begin by focusing on the expression language in chapter 3. Chapters 4 and 5 will be a breeze, but the rest of part 2 should be useful in orienting you to JSTL's tag-set. The examples in part 3 will be useful, and part 4 is specially intended for you. Also, appendix B is both a thorough reference and an indispensable introduction to some of JSTL's advanced features.

If you have a background in JSP but aren't familiar with Java, then parts 2 and 3 will be particularly useful to you. Also, the beginning of chapter 14—integrating JSTL with scriptlets—might be helpful.

In general, the book gets more advanced as it moves forward. Most readers will gain less by reading the book backward (but if you find any interesting hidden messages that way, be sure to let me know).

## Conventions

---

By and large, the book is self-explanatory. I've followed a few conventions throughout the book that should help illuminate some material; a general convention suggests that I list them here. They include:

- *Boldface type*

In code listings, I use **boldface type** to differentiate dynamic code (JSTL tags) from static text (including HTML tags). This distinction is useful because they look the same on the surface, so they can easily blur, especially late at night. Also, a few examples use a JSTL tag *within* an HTML tag, and boldface is helpful to make sure the JSTL tag stands out.

- *Other type styles*

I occasionally use *italics* when introducing a term I want you to remember—or a word that I'd accompany with a bang on the table if I were speaking to you in person (and if there were a table present). `Courier` font marks tag

names (for HTML, XML, or JSTL tags), tag attributes, scoped-variable names, and other words that normally appear within code.

- *Tables for tags*

Just like HTML tags, JSTL tags have *attributes* that let you modify the tags' behavior. For instance, in the tag `<fmt:formatNumber type="currency"/>`, the text `type="currency"` is an attribute. I've listed tag attributes in tables that have a consistent format. Here's an example:

`<c:spam>` tag attributes



Attribute	Description	Required	Default
email	Email address to send junk email to	Yes	<i>None</i>
subject	Subject of the junk-email message	No	"See me and my friends naked."
message	Body of the junk-email message	No	<i>Body</i>

This sample table shows a few things. First, tables for tag attributes have a “tag” icon to help you find them. Such tables have four columns describing the attribute name, a brief description of each attribute, information about whether the attribute must be specified for each use of the tag, and information about the default value of the attribute if you don't specify a value. If the Default column contains *None*, the attribute has no default. If this column contains *Body*, the default value comes from the tag's body. (See chapter 2 for more information about tags, attributes, and bodies. Note that `<c:spam>` is, of course, not a real JSTL tag—although given the number of applications that send out junk mail, there's clearly a need for it; perhaps we'll see it in JSTL 1.1.)

- *Highlighting*

I highlight sections of code samples whenever I feel like it, usually to draw your attention to a part of the code sample that has changed. Highlighting isn't consistent; it's there only when I think it will be useful.

- *Code annotations*

Some longer examples are annotated using bullets like this: ⑧. These are often tied to paragraphs that follow and amplify the code.

- *Call-out boxes*

Occasionally, I draw your attention to a Note, Tip, or Warning using a noticeable box in the middle of the page. To be honest, I do this just because other books do it; fortunately, I use these boxes sparingly.

## **Source code**

---

All of the source code is downloadable from <http://www.manning.com/bayern>. I typed it all in so that you don't have to. Don't thank me too much, because I had to type it into the manuscript anyway. Visit <http://www.manning.com/bayern> to download the code. It's available in a number of convenient formats, including a ready-to-use bundle that can get you up and running quickly, even if you haven't yet set up a JSP container.

## **Author online**

---

I spend a lot of time online, and now, having written this book, I'm eager to discuss it with you and answer any questions you have about it.

Manning has set up an Author Online forum for *JSTL in Action* to make it easy for you to communicate with me and other readers. The Author Online forum is great if you have any questions or comments about the book (or even if you just want to hold me accountable for one of my jokes). To access the Author Online forum, visit <http://www.manning.com/bayern>. This page will help you register, read other people's messages, and post your own questions and answers.