**Creating Custom Tags in JSP: Advantages, Disadvantages, Requirements, and Reflections**

Megan Wheeler

CSD 430: Server Side Development

Instructor: Darrell Payne

16 July 2025

**Abstract**

This paper explores the use of custom tags in JavaServer Pages (JSP), focusing on their advantages, disadvantages, and the requirements for their development. Custom tags are a powerful feature that enables developers to encapsulate reusable logic, promote code reuse, and maintain a clear separation between business logic and presentation. The discussion includes a personal reflection on the practical value of custom tags, supported by examples and APA-style references. The paper concludes that, while custom tags introduce some complexity, their benefits for large-scale web applications far outweigh the drawbacks.

**Creating Custom Tags in JSP: Advantages, Disadvantages, Requirements, and Reflections**

**Introduction**

JavaServer Pages (JSP) custom tags are user-defined elements that extend the capabilities of JSP by encapsulating reusable logic into tag-like syntax. These tags are implemented in Java and can be used to simplify JSP pages, promote code reuse, and separate business logic from presentation. This paper explores the advantages and disadvantages of custom tags, outlines the requirements for developing them, and provides a personal perspective on their use, supported by examples and APA-style references.

**Advantages of Custom Tags**

Custom tags offer several significant benefits in JSP development:

1. **Separation of Concerns:** Custom tags allow developers to move business logic out of JSP pages and into Java classes (tag handlers), making the pages cleaner and easier to maintain. As noted by GeeksforGeeks, “Custom tags separate the business logic from Java Server Pages. JSP still controls the flow but processing is delegated to a separate java class (tag handler class)” (GeeksforGeeks, 2025).
2. **Reusability:** Once created, custom tags can be reused across multiple JSP pages and even different projects, reducing code duplication and promoting consistency (Java Guides, 2019).
3. **Readability and Maintainability:** By abstracting complex logic into tags, JSP pages become more readable, allowing front-end developers to focus on layout and design without delving into Java code (Oracle, n.d.).
4. **Encapsulation:** Custom tags encapsulate functionality, making it easier to test, debug, and update specific features without affecting the rest of the application.
5. **Support for Attributes and Nesting:** Custom tags can accept attributes and be nested, enabling the creation of sophisticated, modular components (Oracle, n.d.).

**Disadvantages of Custom Tags**

Despite their benefits, custom tags also have some drawbacks:

1. **Learning Curve:** Developing custom tags requires knowledge of Java, JSP, and the tag library descriptor (TLD) format, which can be challenging for beginners (TutorialsPoint, n.d.).
2. **Increased Complexity:** For small projects, the overhead of creating and maintaining custom tags may outweigh their benefits.
3. **Performance Overhead:** While generally efficient, custom tags introduce an extra layer of abstraction, which can have a minor impact on performance compared to direct scriptlets, though this is rarely significant in modern applications (GeeksforGeeks, 2025).
4. **Deployment and Configuration:** Custom tags require proper packaging and configuration (e.g., TLD files, class files in the correct directories), which can complicate deployment.

**Requirements for Developing a Custom Tag**

To develop a custom tag in JSP, the following components are required:

1. **Tag Handler Class:** A Java class that implements the tag’s logic. This class typically extends `SimpleTagSupport` or implements interfaces like `Tag`, `BodyTag`, or `SimpleTag` (GeeksforGeeks, 2025).
2. **Tag Library Descriptor (TLD) File:** An XML file that describes the tag, its attributes, and the handler class. The TLD file is placed in the `WEB-INF` directory.
3. **Taglib Directive in JSP:** The JSP page must include a taglib directive to reference the custom tag library, e.g., `<%@ taglib uri="/WEB-INF/mytags.tld" prefix="my" %>`.

**Personal Reflection**

In my experience, custom tags are invaluable for large-scale JSP applications where code reuse, maintainability, and separation of concerns are priorities. They enable teams to create a library of reusable components, streamline development, and reduce errors. However, for small projects or simple pages, the setup and configuration may be excessive.

I prefer custom tags over scriptlets because they keep JSP pages clean and focused on presentation. Scriptlets, which embed Java code directly in JSP, can quickly lead to tangled, hard-to-maintain code. As GeeksforGeeks (2025) states, “Scriptless JSPs are a lot easier to read and understand as compared to JSPs embedded with scriptlets.”

That said, custom tags do require a solid understanding of both Java and JSP, and the initial learning curve can be steep. For teams with mixed skill levels, it may be necessary to provide training or documentation.

**Conclusion**

Custom tags in JSP are a powerful tool for building maintainable, reusable, and readable web applications. Their advantages in promoting separation of concerns, reusability, and maintainability make them a best practice for enterprise-level development. However, they come with a learning curve and some setup overhead. For most modern JSP projects, the benefits far outweigh the drawbacks, especially when compared to scriptlets.

**References**

GeeksforGeeks. (2025, April 28). Custom tags in JSP. <https://www.geeksforgeeks.org/java/custom-tags-in-jsp/>

GeeksforGeeks. (2025, April 28). Custom tags with body in JSP. <https://www.geeksforgeeks.org/java/custom-tags-with-body-in-jsp/>

Java Guides. (2019, January). Tag library and tag library descriptor (TLD). <https://www.javaguides.net/2019/01/tag-libraries-and-tag-library.html>

Oracle. (n.d.). Custom tags in JSP pages. <https://docs.oracle.com/cd/E19575-01/819-3669/bnalj/index.html>

TutorialsPoint. (n.d.). JSP - custom tags. <https://www.tutorialspoint.com/jsp/jsp_custom_tags.htm>