**JSPs: Dynamic vs. Static**

**What**  
Static pages are just plain HTML and CSS. They’re simple, but the content is always the same no matter who visits. JSPs (JavaServer Pages), on the other hand, are dynamic. They run on the server, and the HTML you see can change depending on user input, database results, or even the time of day. JSPs came out of the early Java web stack as a way to mix Java code with HTML, and they’re still part of the Jakarta EE platform today.

**How**  
I see static pages as good for things like a homepage banner or a “Contact Us” section that never changes. Dynamic JSPs are better for anything interactive—like showing someone’s name after they log in, or displaying data from a form. When working with JSP, I like keeping HTML clean and pushing style into a separate CSS file. It keeps the code organized, and it’s easier to make changes later without messing with the Java logic.

**Why**  
Most real applications need to respond to users in some way. Static pages are fine for basics, but JSPs make the site feel alive and responsive. By combining JSP with external CSS, you get the best of both worlds—Java handles the logic, and CSS handles the design. That separation makes teamwork and future updates a lot smoother.

**Quick Example**

<!-- form.jsp -->

<link rel="stylesheet" href="styles.css">

<form action="greet.jsp" method="get">

<label>Enter your name: <input name="username"></label>

<button type="submit">Submit</button>

</form>

<!-- greet.jsp -->

<%@ page contentType="text/html;charset=UTF-8" %>

<link rel="stylesheet" href="styles.css">

<h2>Hello, <%= request.getParameter("username") %>!</h2>

/\* styles.css \*/

body { font-family: Arial, sans-serif; }

h2 { color: darkblue; }

**References**

GeeksforGeeks. (2024). *Introduction to JSP*. <https://www.geeksforgeeks.org/introduction-to-jsp/>  
TutorialsPoint. (2024). *JSP—Expression Language (EL)*. <https://www.tutorialspoint.com/jsp/jsp_expression_language.htm>