

# JEE Web Technologies Comparison

Absolutely! Here's a detailed comparison between **JSP** and **JSTL**, now with **real examples** to help you visualize the differences clearly:

## vs JSP vs JSTL – With Examples

Feature	JSP	JSTL
<b>Purpose</b>	Build structure/content, allows Java code	Perform logic using tags (cleaner, no Java)
<b>Type</b>	View templating	Tag library inside JSP
<b>Syntax</b>	Uses HTML + can embed Java	Uses tag-based XML-like syntax
<b>Usage</b>	HTML layout + can contain loops/logic (not ideal)	Conditions, loops, formatting, etc.
<b>Maintainability</b>	Can be messy if it has lots of Java code	Cleaner and easier to read

### ♦ 1. Display Data (Simple)

#### ✓ JSP:

```
jsp

<%
    String name = "Alice";
    out.print("Hello, " + name);
%>
```

#### ✓ JSTL:

```
jsp

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:set var="name" value="Alice"/>
<p>Hello, ${name}</p>
```

👉 JSTL + EL is much cleaner than using `out.print`.

### ♦ 2. Conditional Display (if-else)

#### ✗ JSP:

```
jsp

<%
    int age = 18;
    if (age >= 18) {
%>
        <p>You are an adult.</p>
<%
    } else {
%>
```

```
<p>You are a minor.</p>

<%
}
%>
```

### ✓ JSTL:

```
jsp

<c:set var="age" value="18" />
<c:if test="${age} >= 18">
    <p>You are an adult.</p>
</c:if>
<c:if test="${age} < 18">
    <p>You are a minor.</p>
</c:if>
```

 JSTL keeps logic **out of Java**, more readable and easier to maintain.

---

## ♦ 3. Loop Through List (for-each)

### ✗ JSP:

```
jsp

<%
    List<String> items = Arrays.asList("Apple", "Banana", "Cherry");
    request.setAttribute("items", items);
%>
<ul>
<%
    for(String item : items) {
%>
    <li><%= item %></li>
<%
    }
%>
</ul>
```

### ✓ JSTL:

(Assume the list `items` is already set as a request attribute by a servlet)

```
jsp

<ul>
<c:forEach var="item" items="${items}">
    <li>${item}</li>
</c:forEach>
</ul>
```

 JSTL `forEach` is declarative and avoids cluttered Java syntax in JSP.

---

## ♦ 4. Formatting (e.g., numbers or dates)

### ✗ JSP:

```
jsp

<%
    double price = 12345.678;
    java.text.DecimalFormat df = new java.text.DecimalFormat("#,###.00");
%>
<p>Price: <%= df.format(price) %></p>
```

## ✓ JSTL:

```
jsp

<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<fmt:formatNumber value="12345.678" type="number" pattern="#,###.00"/>
```

 JSTL `fmt` tag handles formatting without manual Java code.

## ✓ Summary Table with Examples:

Aspect	JSP Example	JSTL Equivalent
Print value	`<%= name %>`	`\${name}`
Set variable	`<% String x = "Hi"; %>`	`<c:set var="x" value="Hi"/>`
If condition	`if(age > 18)`	`<c:if test="\${age > 18}">`
Loop	`for(String x : list)`	`<c:forEach var="x" items="\${list}">`
Format number	`DecimalFormat df = ...`	`<fmt:formatNumber ... />`

Would you like me to generate a mini sample project using Servlets + JSP + JSTL to demonstrate all of this in a working structure?