**JSP Tags**

JSP scripting language include several tags or scripting elements that performs various tasks such as declaring variables and methods, writing expressions, and calling other JSP pages. These are known as JSP scripting elements. The different types of scripting elements are summarized below:

1. **Scripting Elements**

The scripting elements provides the ability to insert java code inside the jsp.

1. **Directive Tags**

Directive tags provide global information about the JSP page and its attributes.

1. **Action Tags**

Action tags are used to control the behavior of the servlet engine. Examples include:

**4. Expression Language (EL)**

Expression Language provides a simplified way to access data stored in JavaBeans, implicit objects, or collections, and is denoted by ${}.

**Scripting Elements**

1. **Declarations (<%! ... %>)**

Used to declare variables or methods that will be available throughout the JSP page.

Code written inside declarations is placed outside the service() method, meaning it's like writing class-level code (variables, methods, etc.).

<%! code %>

<%!

int counter = 0;

public String greetUser(String name) {

return "Hello, " + name;

}

%>

1. **Scriptlets (<% ... %>)**

* Contains Java code that is executed each time the JSP page is requested.
* Code inside scriptlets is placed inside the service() method of the generated servlet.

<% code %>

<%

counter++;

out.println("The counter is: " + counter);

%>

1. **Expressions (<%= ... %>)**

* Used to directly output the result of an expression to the client's browser.
* Whatever you put inside <%= %> is evaluated and automatically sent to the output stream.

<%= expression %>

<%= greetUser("John") %>

**Summary of Differences:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scripting Tag** | **Usage** | **Placement** | **Example** |
| **Declaration** (<%! ... %>) | Declares variables or methods | Class-level (outside service() method) | <%! int x = 0; %> |
| **Scriptlet** (<% ... %>) | Contains Java logic that is executed | Inside service() method | <% x++; %> |
| **Expression** (<%= ... %>) | Outputs the result of an expression | Directly in the HTML output | <%= x %> |

**directive tags**

1. **Page Directive**

The page directive provides information about the current JSP page, such as the language used, importing classes, error handling, session management, etc.

<%@ page attribute="value" %>

**language**: Specifies the scripting language used in the JSP page (default is "java").

<%@ page language="java" %>

**import**: Imports Java classes or packages (similar to import in Java).

<%@ page import="java.util.\*, java.io.\*" %>

**contentType**: Sets the MIME type and character encoding of the response.

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

**errorPage**: Specifies a page to handle exceptions (used in case of errors).

<%@ page errorPage="error.jsp" %>

**isErrorPage**: Used to indicate that the current page is an error page (to access the exception object).

<%@ page isErrorPage="true" %>

**session**: Specifies whether the JSP page uses a session object (default is true).

<%@ page session="false" %>

<%@ page language="java" contentType="text/html;charset=UTF-8" import="java.util.Date" %>

1. **Include Directive**

The include directive is used to include the content of another file (such as another JSP page, HTML file, or text file) during the JSP translation phase (before it’s compiled into a servlet). It is useful for code reuse (e.g., header, footer, etc.).

<%@ include file="relativeURL" %>

<%@ include file="header.jsp" %>

<%@ include file="footer.jsp" %>

1. **Taglib Directive**

The taglib directive is used to declare a custom tag library in a JSP page. This allows you to use custom JSP tags provided by libraries, such as JSTL (JSP Standard Tag Library) or your own custom tags.

<%@ taglib uri="tagLibraryURI" prefix="prefix" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

**Summary of Differences:**

|  |  |  |
| --- | --- | --- |
| **Directive Type** | **Purpose** | **Syntax** |
| **Page Directive** | Provides global settings for the JSP page | <%@ page attribute="value" %> |
| **Include Directive** | Includes a file at JSP compile time | <%@ include file="relativeURL" %> |
| **Taglib Directive** | Declares a tag library to use custom tags | <%@ taglib uri="URI" prefix="prefix" %> |

**action tags**

In JSP (JavaServer Pages), **action tags** are used to control the behavior of the JSP page, interact with JavaBeans, forward requests, include resources, and perform other operations. Action tags are predefined XML-like tags that provide functionality at runtime, unlike directive tags, which are processed at translation time.

1. **<jsp:include>**

The <jsp:include> action tag is used to include the content of another resource (like another JSP file, HTML file, or Servlet) **dynamically** at runtime. The included file is processed each time the request is made.

<jsp:include page="relativeURL" flush="true" />

<jsp:include page="header.jsp" />

<jsp:include page="footer.jsp" />

1. **<jsp:forward>**

The <jsp:forward> action tag is used to **forward the request** to another resource (such as another JSP, servlet, or HTML file). After the forward, the control is passed to the new resource, and the original page stops processing.

<jsp:forward page="relativeURL" />

<jsp:forward page="success.jsp" />

1. **<jsp:param>**

The <jsp:param> action tag is used in conjunction with <jsp:include> or <jsp:forward> to pass parameters to the included or forwarded resource.

<jsp:param name="paramName" value="paramValue" />

<jsp:forward page="success.jsp">

<jsp:param name="username" value="John" />

<jsp:param name="role" value="admin" />

</jsp:forward>

1. **<jsp:useBean>**

The <jsp:useBean> action tag is used to instantiate or locate a **JavaBean** and make it available in the JSP page. It can either create a new bean object or reference an existing one based on the specified scope.

<jsp:useBean id="beanName" class="fullyQualifiedClassName" scope="scope" />

<jsp:useBean id="user" class="com.example.UserBean" scope="session" />

1. **<jsp:setProperty>**

The <jsp:setProperty> action tag is used to set a property value in a JavaBean, either from a request parameter or a specified value.

<jsp:setProperty name="beanName" property="propertyName" value="value" />

<jsp:setProperty name="user" property="username" value="John" />

<jsp:setProperty name="user" property="age" value="30" />

<jsp:setProperty name="user" property="\*" />

1. **<jsp:getProperty>**

The <jsp:getProperty> action tag is used to retrieve the value of a property from a JavaBean and output it to the client.

<jsp:getProperty name="beanName" property="propertyName" />

<jsp:getProperty name="user" property="username" />

1. **<jsp:plugin>**

The <jsp:plugin> action tag is used to **embed a Java applet or JavaBean** in the generated HTML page, specifying how to handle the browser plugin needed to run it.

<jsp:plugin type="bean|applet" code="className" codebase="URL" width="width" height="height">

<jsp:param name="paramName" value="paramValue" />

</jsp:plugin>

<jsp:plugin type="applet" code="MyApplet.class" codebase="appletPath" width="300" height="200">

<jsp:param name="param1" value="value1" />

</jsp:plugin>

**Summary of Differences:**

|  |  |  |
| --- | --- | --- |
| **Action Tag** | **Description** | **Syntax Example** |
| **<jsp:include>** | Dynamically includes another resource during request processing. | <jsp:include page="header.jsp" flush="true" /> |
| **<jsp:forward>** | Forwards the current request to another resource. | <jsp:forward page="success.jsp" /> |
| **<jsp:param>** | Passes parameters to the included or forwarded resource. | <jsp:param name="username" value="John" /> |
| **<jsp:useBean>** | Instantiates or finds a JavaBean and makes it available for use in the JSP page. | <jsp:useBean id="user" class="com.example.UserBean" scope="session" /> |
| **<jsp:setProperty>** | Sets properties in a JavaBean, either from request parameters or specified values. | <jsp:setProperty name="user" property="username" value="John" /> |
| **<jsp:getProperty>** | Retrieves the value of a JavaBean property and displays it. | <jsp:getProperty name="user" property="username" /> |
| **<jsp:plugin>** | Embeds a Java applet or JavaBean in the page and ensures the required plugin is available. | <jsp:plugin type="applet" code="MyApplet.class" width="300" height="200" /> |

**Expression Language (EL)**

JSP **Expression Language (EL)** provides a way to access data stored in JavaBeans, request attributes, session attributes, and application attributes in JSP pages without needing to write Java code. EL simplifies the syntax for accessing and manipulating data in JSP, allowing more readable and concise expressions.

EL is primarily used for retrieving data but can also perform basic operations such as arithmetic, logical, relational, and conditional expressions.

**EL Types:**

1. **Value Expressions**
2. **Method Expressions**

**1. Value Expressions**

Value expressions in EL are used to access or modify data (usually from JSP implicit objects like request, session, application, etc.). These are further categorized into two types:

**a. Immediate Value Expressions**

* Used to **retrieve** values.
* Denoted by the syntax ${expression}.

<p>Username: ${user.username}</p>

<h:outputText value="#{user.username}" />

|  |  |  |
| --- | --- | --- |
| **Implicit Object** | **Description** | **Example** |
| **pageScope** | Attributes available in the current page scope | ${pageScope.myVar} |
| **requestScope** | Attributes available in the request scope | ${requestScope.name} |
| **sessionScope** | Attributes available in the session scope | ${sessionScope.user} |
| **applicationScope** | Attributes available in the application scope | ${applicationScope.data} |
| **param** | Request parameters (String values) | ${param.username} |
| **paramValues** | Request parameters (String array) | ${paramValues.interests} |
| **header** | HTTP request headers | ${header["User-Agent"]} |