

## **Directive Elements :**

JSP directives are the elements of a JSP source code that guide the web container on how to translate the JSP page into its respective servlet.

### **Syntax :**

@

```
<% @ directive attribute = "value"%>
```

Directives can have a number of attributes which you can list down as **key-value pairs** and separated by commas. The blanks between the @ symbol and the directive name, and between the last attribute and the closing %>, are optional.

### **Different types of JSP directives :**

There are three different JSP directives available. They are as follows:

- **Page Directives :** JSP page directive is used to define the properties applying the JSP page, such as the size of the allocated buffer, imported packages and classes/interfaces, defining what type of page it is etc. The syntax of JSP page directive is as follows:

```
<% @page attribute = "value"%>
```

- **Different properties/attributes :**

The following are the different properties that can be defined using page directive :

- **import:** This tells the container what packages/classes are needed to be imported into the program.

#### **Syntax:**

```
<% @page import = "value"%>
```

- **Example :**

- html

```
<%-- JSP code to demonstrate how to use page  
directive to import a package --%>
```

```
<% @page import = "java.util.Date"%>
```

```
<%Date d = new Date();%>
```

```
<%=d%>
```

## **Creating Template Text**

In most cases, a large percentage of your JSP document consists of static text (usually HTML), known as template text . In almost all respects, this HTML looks just like normal HTML, follows all the same syntax rules, and is simply "passed through" to the client by the servlet created to handle the page. Not only does the HTML look normal, it can be created by whatever tools you already are using for building Web pages. For example, we used Macromedia Dreamweaver for many of the JSP pages in this book.

There are two minor exceptions to the "template text is passed straight through" rule. First, if you want to have <% or %> in the output, you need to put <\<% or %\> in the template text. Second, if you want a comment to appear in the JSP page but not in the resultant document, use

<%-- JSP Comment --%>  
HTML comments of the form

<!-- HTML Comment -->  
are passed through to the client normally.

The **<jsp:text>** action can be used to write the template text in JSP pages and documents. Following is the simple syntax for this action –

```
<jsp:text>Template data</jsp:text>
```

### Action Elements:

**JSP** actions can be used to print a script expression, create and store a Java Bean and for many other things. They are used for constructs in XML syntax to control the behavior of the **Servlet engine**.

### Using JSP actions

- a file can be inserted into another page dynamically,
- a bean component can be reused,
- a user can be forwarded from one page to another page.

**Following are the JSP action tags:**

Action tags	Description
<jsp:include>	Includes a file at the time when the page is requested.
<jsp:useBean>	Finds the object of Java bean from given scope or create a new object.
<jsp:setProperty>	Sets the property of a JavaBean object.
<jsp:getProperty>	Prints the property of the Java bean object.
<jsp:forward>	Forwards the request and response to another resource.
<jsp:plugin>	It is used when there is a need of a plugin to run a Bean class.
<jsp:param>	Sets the parameter value to the request. It is used in forward and include mostly.
<jsp:fallback>	Supplies alternate text if Java applet is unavailable on the client.
<jsp:element>	Defines XML element directly.

## The <jsp:include> Action Tag

### Syntax:

```
<jsp:include page="page URL" flush="Boolean Value" />
```

Example : Illustrating the <jsp:include> tag

```
//welcome.jsp
```

```
<p>  
    Today's date: <%= new java.util.Date()%>  
</p>
```

```
//demo.jsp
```

```
<html>  
    <head>  
        <title>JSP include Action Tag</title>  
    </head>  
<body>  
    <h3>JSP page: Demo Include</h3>  
    <jsp:include page = "welcome.jsp" flush="false" />  
</body>  
</html>
```

## The <jsp:forward> Action Tag

### Syntax:

```
<jsp:forward page="Relative URL" />
```

Example : Illustrating the <jsp:forward> action tag

```
<html>  
    <head>  
        <title>JSP Forward Action Tag</title>  
    </head>  
<body>  
    <h2>JSP page: Demo forward</h2>
```

```
<jsp:forward page="welcome.jsp" />  
</body>  
</html>
```