1. **Directives**  
   In the directives we can import packages, define error handling pages or the session information of the JSP page.
2. **Declarations**  
   This tag is used for defining the functions and variables to be used in the JSP.
3. **Scriplets**  
   In this tag we can insert any amount of valid java code and these codes are placed in \_*jspService*method by the JSP engine.
4. **Expressions**  
   We can use this tag to output any data on the generated page. These data are automatically converted to string and printed on the output stream.

**DIRECTIVES**

Syntax of JSP directives is:

<%@directive attribute="value" %>

Where directive may be:

1. page: page is used to provide the information about it.  
   Example: <%@page language="java" %>
2. include: include is used to include a file in the JSP page.  
   Example: <%@ include file="/header.jsp" %>
3. taglib: taglib is used to use the custom tags in the JSP pages (custom tags allows us to defined our own tags).  
   Example: <%@ taglib uri="tlds/taglib.tld" prefix="mytag" %>

and attribute may be:

1. language="java"  
   This tells the server that the page is using the java language. Current JSP specification supports only java language.  
   Example: <%@page language="java" %>
2. extends="mypackage.myclass"  
   This attribute is used when we want to extend any class. We can use comma(,) to import more than one packages.  
   Example: <%@page language="java" import="java.sql.\*,mypackage.myclass" %>
3. session="true"  
   When this value is true session data is available to the JSP page otherwise not. By default this value is true.  
   Example: <%@page language="java" session="true" %>
4. errorPage="error.jsp"  
   errorPage is used to handle the un-handled exceptions in the page.  
   Example: <%@page language="java" session="true" errorPage="error.jsp"  %>
5. contentType="text/html;charset=ISO-8859-1"  
   Use this attribute to set the mime type and character set of the JSP.  
   Example: <%@page language="java" session="true" contentType="text/html;charset=ISO-8859-1"  %>

**Declaration**

Declaration in JSP is way to define global java variable and method. This java variable method in declaration can be access normally. Normally declaration does not produce any output, and access of this code is also limited. The declaration code does not reside inside service method of JSP. Declaration we use method to convert null convert of string in JSP.

**Example of Declaration in JSP (declaration.jsp)**

<%@ page language="java" errorPage="" %>

<%!

**public** **String** nullconv(**String** str)

{

**if**(str==**null**)

str="";

**else** **if**(str.equals("null"))

str="";

**else** **if**((str.trim()).equals(""))

str="";

**else** **if**(str.equals(**null**))

str="";

**else**

str=str.trim();

**return** str;

}

%>

**Scriplets :**

Syntax of JSP Scriptles are:

  <%  
  //java codes  
   %>

JSP Scriptlets begins with <% and ends %> .We can embed any amount of java code in the JSP Scriptlets. JSP Engine places these code in the *\_jspService()*method. Variables available to the JSP Scriptlets are:

* request:  
  request represents the clients request and is a subclass of *HttpServletRequest.*Use this variable to retrieve the data submitted along the request.  
  Example:  
    <%  
    //java codes  
     String userName=null;  
     userName=request.getParameter("userName");  
     %>
* response:  
  response is subclass of *HttpServletResponse*.
* session:  
  session represents the HTTP session object associated with the request.
* out:  
  out is an object of output stream and is used to send any output to the client.

Other variable available to the scriptlets are *pageContext, application,config*and*exception.*

**Expression**

Expression tag ( <%=   %>)

This tag allows the developer to embed any Java expression and is short for out.println().

A semicolon ( ; ) does not appear at the end of the code inside the tag.

For example,to show the current date and time.

Date : <%= new java.util.Date() %>