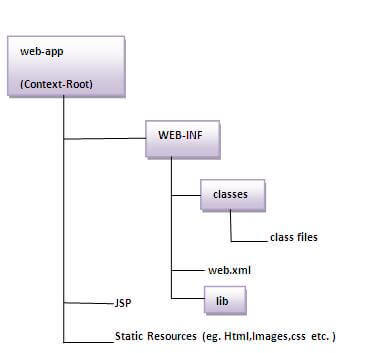
### **Directory structure of JSP**

The directory structure of JSP page is same as Servlet. We contain the JSP page outside the WEB-INF folder or in any directory.



The JSP pages follow these phases:

**TCC,II RD**

* Translation of JSP Page
* Compilation of JSP Page
* Classloading (the classloader loads class file)
* Instantiation (Object of the Generated Servlet is created).
* Initialization ( the container invokes jspInit() method).
* Request processing ( the container invokes \_jspService() method).
* Destroy ( the container invokes jspDestroy() method).

#### **Note: jspInit(), \_jspService() and jspDestroy() are the lif**

## ****JSP Scripting Elements:****

The scripting elements provide the ability to **insert java code inside the JSP.** There are three types of scripting elements:

* **scriptlet tag**–  A scriptlet tag is used to execute Java source code in JSP.  
  Syntax is: **<%  java source code %>**
* **expression tag**– The code placed within JSP expression tag is written to the output stream of the response. So you need not write **out.print()** to write data. It is mainly used to print the values of [variable](https://www.edureka.co/blog/java-tutorial/#variables) or method.  
  Syntax : **<%=  statement %>**
* **declaration tag**– The JSP declaration tag is used to declare fields and methods. The code written inside the JSP declaration tag is placed **outside the service()** method of an auto-generated servlet. So it doesn’t get memory at each request.  
  Syntax: **<%!  field or method declaration %>**

# JSP directives:The jsp directives are messages that tells the web container how to translate a JSP page into the corresponding servlet.

# In JSP, directive is described in <%@ %> tags.

There are three types of directives:

* page directive
* include directive
* taglib directive

**page directive:**

The page directive defines attributes that apply to an entire JSP page.

**<%@ page attribute="value" %>**

Example:

<%@ page **import**="java.util.Date" %> //similar to import keyword in java

**<%@ page contentType=application/msword %>**

//type of the HTTP response.**The default value is "text/html;charset=ISO-8859-1**".

**<%@ page errorPage="myerrorpage.jsp" %>**

if exception occurs in the current page, it will be redirected to the error page.

**<%@ page isErrorPage="true" %>  //Exception object use in this**

The isErrorPage attribute is used to declare that the current page is the error page.

### **isThreadSafe**

Servlet and JSP both are multithreaded. Container will wait until the JSP finishes responding to a request before passing another request to it.If you make the value of isThreadSafe attribute like:

<%@ page isThreadSafe="false" %>//**by default true**

**include directive**

**<%@ include file="resourceName" %>**

Example:

**<%@ include file="header.html" %>**

**taglib directive**

We use the TLD (Tag Library Descriptor) file to define the tags.

<%@ taglib uri="uriofthetaglibrary" prefix="prefixoftaglibrary" %>

<%@ taglib uri="http://www.javatpoint.com/tags" prefix="mytag" %>

There are **9 jsp implicit objects**. **These objects are *created by the web container* that are available to all the jsp pages.**

## What is JSP Implicit object?

* JSP implicit objects are created during the translation phase of JSP to the servlet.
* These objects can be directly used in scriplets that goes in the service method.
* They are created by the container automatically, and they can be accessed using objects.

There are 9 types of implicit objects available in the container:

1. out
2. request
3. response
4. **config**: "Config" is of the type java.servlet.servletConfig
5. **application**: Application object (code line 10) is an instance of javax.servlet.ServletContext and it is used to get the context information and attributes in JSP.
6. session
7. pageContext
8. page
9. exception

Example:

1.

<% String servletName = config.getServletName();

out.println("Servlet Name is " +servletName);%>

2.

<%String driver=application.getInitParameter("dname");  %>

3.<% session.setAttribute("user","GuruJSP"); %>

4.<% String pageName = page.toString();

out.println("Page Name is " +pageName);%>

5.<%= exception %>

**Scopes are of 4 types**:SARP

* **Page**
* Request
* Session
* Application

In JSP, page scope is the **default** **scope**.

There are 11 types of Standard Action Tags as following:

* jsp:useBean
* jsp:include
* jsp:setProperty
* jsp:getProperty
* jsp:forward
* jsp:plugin
* jsp:attribute
* jsp:body
* jsp:text
* jsp:param
* jsp:attribute
* jsp:output

Syn:<jsp:useBean id="" class="" />

Ex:

<jsp:useBean **id**="name" **class**="demotest.DemoClass" **scope**=”optional he..” type=”optional he..”>

<jsp:include page="printdate.jsp" />

<jsp:setproperty name="" property="" >

<jsp:forward page="printdate.jsp" />

### **List the various action tags used in JSP.**

Following are the list of various action tags used in JSP:

* jsp:forward: This action tag forwards the request and response to another resource.
* jsp:include: This action tag is used to include another resource.
* jsp:useBean: This action tag is used to create and locates bean object.
* jsp:setProperty: This action tag is used to set the value of the property of the bean.
* jsp:getProperty: This action tag is used to print the value of the property of the bean.
* jsp:plugin: This action tag is used to **embed another component such as the applet.**
* jsp:param: This action tag is used to set the parameter value. It is used in forward and includes mostly.
* jsp:fallback: This action tag can be used to print the message if the plugin is working.

# Exception Handling in JSP

# <https://www.javatpoint.com/exception-handling-in-jsp>

In JSP, there are two ways to perform exception handling:

In JSP, there are two ways to perform exception handling:

1. By **errorPage** and **isErrorPage** attributes of page directive
2. By **<error-page>** element in web.xml file