**JSP (Java Server Pages)**

JSP is a web component that run on a server.

Java code inside HTML pages.

JSP Life Cycle

* Loading and instantiating
* Initialization
* Request processing
* End of life

You write the .jsp file.

The container TRANSLATE the .jsp file into .java source code for a servlet class. (The container writes the .java file for the servlet the JSP becomes)

The container COMPILE the servlet .java source into .class file.

The container loads the newly generated servlet class.

The container instantiates the servlet and causes the servlet jspInit() method to run.

The object is now fully fledged servlet ready to accept client requests.

The container creates a new thread to handle a client request and the servlet’s \_jspService() method runs.

Translation and compilation happens only once.

1. jspInit() method.

This method is called from the init method, can override this method.

1. jspDestroy() method.

This method is called from the servlet’s destroy method, can override this method.

1. \_jspService() method.

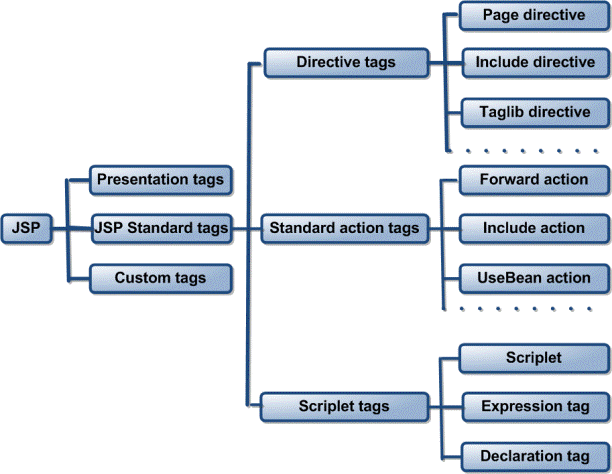
This method is called from servlets service() method and cannot be overridden.

JSP Elements

* Directive elements
* Scripting elements
* Action elements
* Directive Elements

Directive elements provide information to the JSP container about the page.

There are three directives available: page, include, and taglib.



<%@ page attributes %>

<%@ include attributes %>

* Scripting elements
  + <%! declaration %>
  + <% scriptlet code %>
  + <%= expression %>

Declarations is used to declare, and optionally define, a Java variable or a method.

Scriptlets contain Java code statements.

Expressions are used to output the value of a Java expression to the client.

* Action Elements (Standard Actions)

Standard actions are defined by the JSP specification

<jsp:useBean>

<jsp:setProperty>

<jsp:getProperty>

<jsp:param>

<jsp:include>

<jsp:forward>

Implicit Objects

* request, response, out, session, config, exception, application
* Implicit objects are used within scriptlet and expression elements.

Scope

* page – Page scope is the most restrictive. With page scope, the object is only accessible within the page in which it is defined.
* request – With request scope, objects are available for the life of the specific request. This means that the object is available within the page in which it is created, and within pages to which the request is forwarded or included.

Only the execution thread for a particular request can access these objects.

* session – Objects with session scope are available to all application components that participate in the client’s session.
* application – This is the least restrictive scope. Objects that are created with application scope are available to the entire application for the life of the application.