

JSP

(Java Server Pages)

➤ **JSP - Introduction**

- ✓ **JSP provides server side executable content in a web page.**
- ✓ **JSP is a text based document capable of returning both static and dynamic content to a client browser.**
- ✓ **In JSP static and dynamic content can be interchanged.**



JSP VS Servlet

- ✓ **Servlet returns dynamic content to the client machine in form of a web page. JSP do the same.**
- ✓ **Servlet is a java only program and the HTML code has to be embedded within the Java code.**

In JSP all the HTML tags have to be written inside out.println statements; so code become quite large.

➤ Advantages of JSP over Servlet

- ✓ It is difficult to use `println()` statement to generate the HTML page in a servlet
- ✓ No need for compiling, packaging, CLASSPATH setting for JSP
- ✓ Visual content and processing logic are separated.
- ✓ Automatic Deployment : Recompile is done automatically when changes are made to JSP page

➤ Relation between JSP & Servlet

- ✓ Each JSP page is compiled into a servlet before it can be used.
- ✓ This is normally done when the first request to the JSP page is made.

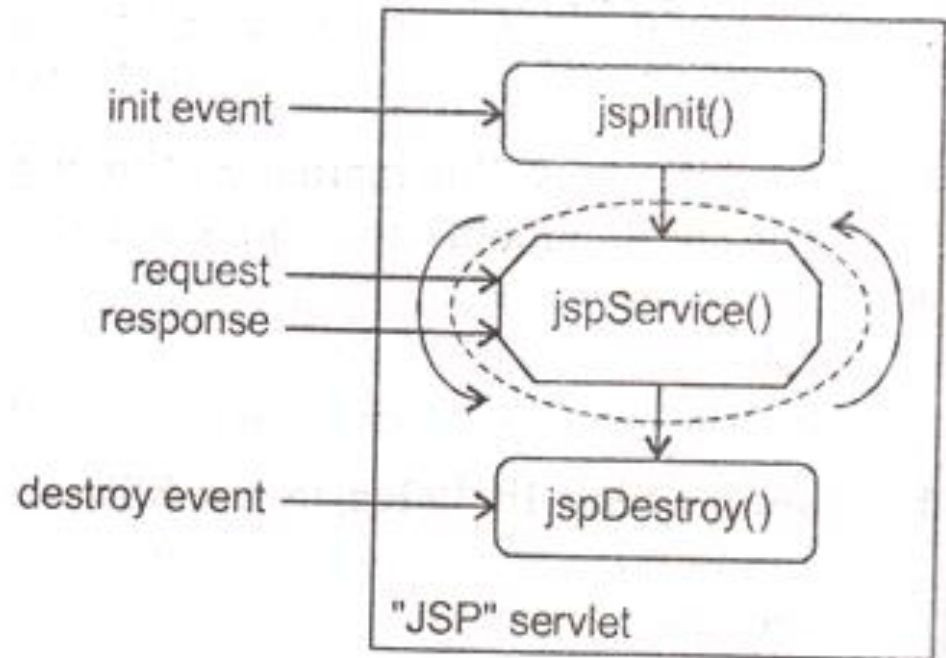
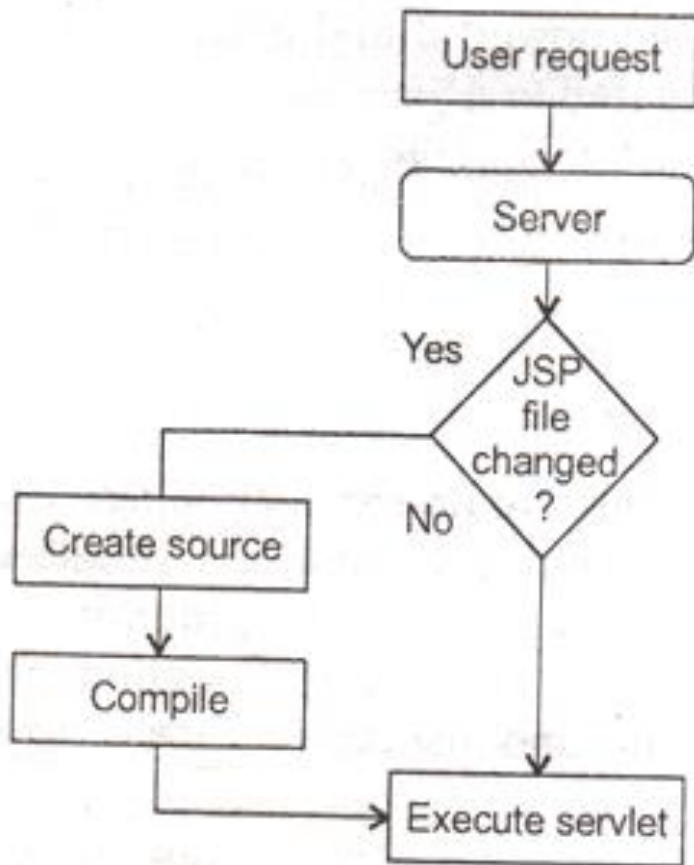




JSP Lifecycle

- ✓ **Translation Phase** : JSP page get translated to servlet code.
- ✓ **Compilation Phase** : Page is requested for first time/modified the servlet code is compiled
- ✓ **Execution Phase** : final phase where jsp servlet methods are executed. Methods are `jspInit()`, `jspService()`, `jspDestory()`

➤ JSP Execution Process and Lifecycle





JSP Server

- ✓ JSP Pages are stored in Web Server so they can be accessed by client =>
- ✓ **Step-1** : Web browser sends a HTTP Req to web server.
- ✓ **Step-2** : Web server compiles the JSP page, if it is requested first time.
- ✓ **Step-3** : Web server executes the compiled version of the requested JSP Page
- ✓ **Step-4** : Web server generates the output of the Execution into HTTP Response.
- ✓ **Step-5** : Web server sends the HTTP Response to the Web Browser.

➤ Implicit Objects in JSP

Implicit Object	Description
config	javax.servlet.ServletConfig getinitparameter() method is used to access the initialization parameters
exception	java.lang.Throwable object that represents an exception that is passes to JSP Page.
out	javax.servlet.jsp.JspWriter object that writes text as part of the response to a request
page	An Object that represents the current JSP instance
request	javax.sevlet.ServletRequest
response	javax.sevlet.ServletResponse
session	javax.servlet.http.HttpSession object that represents the client session information



JSP Elements

- ✓ **1. Directives** : Directives are messages to the JSP Container i.e. the server program that executes JSP. Directives enable to specify custom tag libraries for use in JSP.
- ✓ **2. Scripting Elements** : Scripting elements enable programmers to insert Java code which will be a part of the servlet.
- ✓ **3. Actions** : Actions encapsulates functionality in predefined tags that programmers can embed in a JSP.



1. Directive

✓ **a) Page Directive** : The page directive specifies global settings for the JSP. There can be many directives, provided that there is only one occurrence of each attribute.

Syntax : `<%@ page`

`[language = "java"]`

`[extends = "Class_Name"]`

`[import = "Class_Name"]`

`[session = "true | false"]`

`[buffer = "none | size in kb"]`

`[autoFlush = "true | false"]`

`[errorPage = "URL"]`

`[contentType = "text/html"]`

`%>`



1. Directive

- ✓ **b) Include Directive**: The include directive inserts a file of text or code on a JSP file at translation time, when the JSP file is compiled.
- ✓ **Syntax :**
`<%@ include file="URL" %>`
- ✓ **Ex :**
`<%@ include file="date.jsp" %>`



1. Directive

✓ **c) Taglib Directive**: The taglib directive declares that the JSP file uses custom tags, name the tag library that defines them, and specify their tag prefix.

✓ **Syntax :**

`<%@ taglib uri="URL" prefix="tagPrefix" %>`

✓ **Ex :**

`<%@ taglib uri="/mylib.tld" prefix="my" %>`



2. Scripting Element

✓ **a) Declarations :**

e.g. `<%! int i=0; %>`

✓ **b) Expressions :**

e.g. `<%= Math.sqrt(5) %>`

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

✓ **c) Scriptlet :** The scriptlet contains a set of java statements to be executed as part of the service method of the Servlet class.

Syntax : `<% list of statements %>`

✓ **d) Comments :**

e.g. `<%-- text --%>`