Java Server Pages (JSP)

- It stands for Java Server Pages.
- It is a server side technology.
- It is used for creating web application.
- It is used to create dynamic web content.
- In this JSP tags are used to insert JAVA code into HTML pages.
- It is an advanced version of Servlet Technology.
- It is a Web based technology helps us to create dynamic and platform-independent web pages.
- In this, Java code can be inserted in HTML/ XML pages or both.
- JSP is first converted into servlet by JSP container before processing the client's request.

Servlet	JSP
Servlet is a java code.	JSP is a HTML-based compilation code.
Writing code for servlet is harder than JSP as it is HTML in java.	JSP is easy to code as it is java in HTML.
Servlet plays a controller role in the, MVC approach.	JSP is the view in the MVC approach for showing output.
Servlet is faster than JSP.	JSP is slower than Servlet because the first step in the JSP lifecycle is the translation of JSP to java code and then compile.
Servlet can accept all protocol requests.	JSP only accepts HTTP requests.

Servlet	JSP
In Servlet, we can override the service() method.	In JSP, we cannot override its service() method.
Packages are to be imported on the top of the program.	Packages can be imported into the JSP program (i.e, bottom , middleclient-side, or top)
The facility of writing custom tags is not present.	The facility of writing custom tags is present.
Servlets are hosted and executed on Web Servers.	Before the execution, JSP is compiled in Java Servlets and then it has a similar lifecycle as Servlets.

Life Cycle of JSP :-

1. Translation :- JSP -> Servlet
2. Compilation :- Servlet -> .class(Byte Code)
3. Loading & Instantiation :- .class file is loaded on server and object is created.
4. Initialization :- The created object will be initialized by jspInit() method.
5. Request handling :- The client's request will be handled _jspService() method.
6. Destroy :- The created object will be destroyed jspDestroy() method.

JSP Tags

1. Scripting tags :- <% %> A scriptlet tag is used to execute java source code in JSP
2. Directive tags :- <%@ %> 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.
3. Action tags :- <jsp:- /> Action tags are used to perform some specific tasks within the JSP PAGE. It provides a way with java objects, controls flow , and perform actions like forwarding, including, and manipulating session attributes.

Scripting tags

1. Scriptlet tags :- Scriptlet tags are the tags used to insert pure java code inside JSP, and Scriptlets in JSP are the java codes inserted inside <% %> tags.
2. Declaration tags :- Declaration tag is one of the scripting elements in JSP. <%! %>
   This Tag is used for declare the variables. Along with this, Declaration Tag can also declare method and classes.
3. Expression tags :- Expression tag is one of the scripting elements in JSP. Expression Tag in JSP is used for writing your content on the client-side. We can use this tag for displaying information on the client's browser. <%= %>

Directive tags

1. Page Directive :- The page directive is used to provide instructions to the container that pertain to the current JSP page.

   <%@page attribute="value" %>

2. Include Directive :- The include directive is used to include a file during the translation phase.

<%@include file_name="resource" %>

3. **Taglib Directive :-** The taglib directive declares that your JSP page uses a set of custom tags, identifies the location of the library, and provides means for identifying the custom tags in your JSP page.

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

Action tags

JSP Action Tags	Description
jsp:forward	forwards the request and response to another resource.
jsp:include	includes another resource.
jsp:useBean	creates or locates bean object.
jsp:setProperty	sets the value of property in bean object.
jsp:getProperty	prints the value of property of the bean.
jsp:plugin	embeds another components such as applet.
jsp:param	sets the parameter value. It is used in forward and include mostly.
jsp:fallback	can be used to print the message if plugin is working. It is used in jsp:plugin.

1. ### jsp:forward action tag

   The jsp:forward action tag is used to forward the request to another resource it may be jsp, html or another resource.

   <jsp:forward page="relativeURL | <%= expression %>" />

2. ### jsp:include action tag

   The jsp:include action tag is used to include the content of another resource it may be jsp, html or servlet.

   <jsp:include page="relativeURL | <%= expression %>" />

3. jsp:useBean action tag

   The jsp:useBean action tag is used to locate or instantiate a bean class. If bean object of the Bean class is already created, it doesn't create the bean depending on the scope. But if object of bean is not created, it instantiates the bean

   ```
   <jsp:useBean id= "instanceName" scope= "page | request | session | applica
   tion"
   class= "packageName.className" type= "packageName.className"
   beanName="packageName.className | <%= expression >" >
   </jsp:useBean>
   ```

4. jsp:setProperty and jsp:getProperty action tags

   The setProperty and getProperty action tags are used for developing web application with Java Bean. In web devlopment, bean class is mostly used because it is a reusable software component that represents data.

   The jsp:setProperty action tag sets a property value or values in a bean using the setter method.

   ```
   <jsp:setProperty name="instanceOfBean" property= "*"  |
   property="propertyName" param="parameterName"  |
   property="propertyName" value="{ string | <%= expression %>}"
   />
   ```

5. jsp:getProperty action tag

   The jsp:getProperty action tag returns the value of the property.

   Syntax of jsp:getProperty action tag

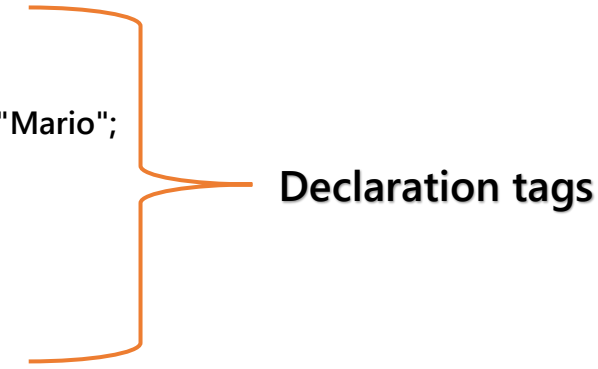
   ```
   <jsp:getProperty name="instanceOfBean" property="propertyN
   ame" />
   ```

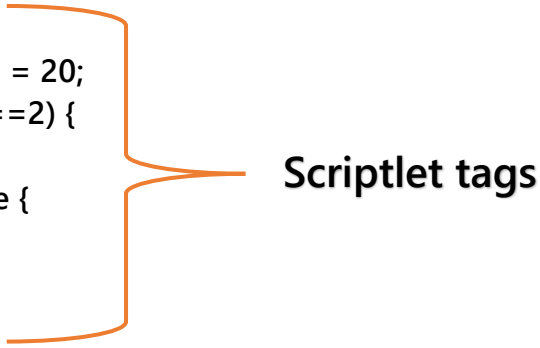
There are many other tags we can use in JSP : -

- JSTL (Java Standard Tag Library)
- Spring MVC Tags
- Java Server Faces (JSF) Tags
- Struts Tags

Scripting tags example :-

```
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

	<%!
		int a = 20;
		String name = "Mario";

		int add() {
			return a*a;
		}
	%>

	<%= name %>
	<%= add() %>

	<%
		int b = 20;
		if(b==2) {

		} else {

		}
	%>
</body>
</html>
```





JSP implicit objects :-

The Java objects that the JSP Container makes available to the developers in each page and the developer can call them directly without being explicitly declared. JSP Implicit Objects are also called pre-defined variables.

request
This is the HttpServletRequest object associated with the request.

response
This is the HttpServletResponse object associated with the response to the client.

out
This is the PrintWriter object used to send output to the client.

session
This is the HttpSession object associated with the request.

application
This is the ServletContext object associated with the application context.

config
This is the ServletConfig object associated with the page.

pageContext
This encapsulates use of server-specific features like higher performance JspWriters.

page
This is simply a synonym for this, and is used to call the methods defined by the translated servlet class.

Exception
The Exception object allows the exception data to be accessed by designated JSP.

Expression Language

The Expression Language (EL) simplifies the accessibility of data stored in the Java Bean component, and other objects like request, session, application etc. There are many implicit objects, operators and reserve words in EL. It is the newly added feature in JSP technology version 2.0.

${ expression }

Some implicit objects in Expression languages are :-

- **requestScope: Contains attributes that are specific to the current HTTP request.**

- **param: Provides access to request parameters sent to the server.**
- **paramValues: Provides access to request parameter values as an array.**
- **sessionScope: Contains attributes that are specific to the current user session.**
- **pageContext: Provides access to various objects and methods related to the JSP page context.**
- **pageScope: Contains attributes that are specific to the current JSP page.**
- **applicationScope: Contains attributes that are specific to the entire web application.**

Etc..

MVC(Model View Controller) Design pattern