# 

# 

# JSP

**Q-1: What is JSP. Why we use it**

**A:** (1)JSP stands for java server page.

(2)JSP is a server side technology like servlet which is used to develop server side web component

(3)JSP will be designed dynamic response for the client.

(4)JSP is a combination of HTML tags nad JAVA code.

(5)the extension for the file must be .jsp .

**Advanteage of jsp->**

JSP offer several advantages as listed below:

1->Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself.

2->JSP are always compiled before it's processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.

3->JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP etc.

4->JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

**Q-2: What is the diff b/w Servlet and JSP.**

**A: Servelet:-**

* We have to write the servlet class by extending HttpServlet or GenericServlet.
* We have to configure the servlet in web.xml or Annotation.
* Servelt lifecycle method are

Public void init(ServletConfig)

Public void service(HttpServletRequest req,HttpServletResponse res)

Public void destroy()

* Servlet will be initialized at container start-up or at first request.
* When you modify the servlet then you have to re-compile ,re-deploy and re-start the server.
* We can define any package for servlet

**JSP:-**

* Container write or generate the Servlet class by extending HttpJspBase.
* No configuration is required for JSP.
* JSP lifecycle method are

Public void jspinit() /\_jspInit()

Public void \_jspservice(HSRequest req,HSResponse res)

Public void jspdestroy() /\_jspdestroy()

* Jsp will be initialize at container start-up or at first request
* When you modify the jsp, Container will do the automatic re-traslation,re-compilation etc.
* Translated servlet will be placed in org.apache,jsp package.

**Q-3: What are the advantages of JSP over JavaScript?**

**A:** JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.

**Q-4: What are the advantages of jsp over servlet?**

The advantage of JSP is that they are document-centric. Servlets, on the other hand, look and act like programs. A Java Server Page can contain Java program fragments that instantiate and execute Java classes, but these occur inside an HTML template file and are primarily used to generate dynamic content.

Some of the JSP functionality can be achieved on the client, using JavaScript. The power of JSP is that it is server-based and provides a framework for Web application development.

**Q-5: What are the LifeCycle Phases of Jsp**

**A:** Whenever the client send the first request to jsp,the container will do the following task.

1. Translate the JSP to Servlet(Show.jsp.java -> show\_jsp.class )
2. Load the translated Servlet Class.
3. Create the instance of translated servlet class
4. Calls the lifecycle method \_jspInit() for initialization.
5. Invoke the lifecycle method \_jspService().

Whenever client sends 2nd onwards request to jsp then \_jspService() will be

Called directly.

1. At the shutdown time container calls the lifecycle method \_jspDestroy().

**Q-6: What are the LifeCycle method of JSP.**

**A:** JSP lifecycle methods are

(1) jspInit() / \_jspInit()

(2) \_jspService()

(3) jspDestroy() / \_jspDestroy()

**Q-7: What jsp life cycle method can I override?**

**A:** We can override jspinit() and jspDestroy() methods but not \_jspService().

**Q-8: Why is \_jspService () start with ‘\_’?**

**A:** \_jspService() method will be written by the container hence any methods which are not to be overridden by the end user are typically written starting with a '\_'. This is the reason why we don't override \_jspService() method in any JSP page.

**Q-9: What is the default super class of Translated Servlet.**

**A:** HttpJspBase

**Q-10: What is the diff b/w jspInit() and \_ jspInit() in jsp.**

**A:** in Tomcat 5,when container is writing translated servlet then it is placing only \_jspService() method inside the translated servlet.so you can override both jspInit() and \_jspInit() methods. in Tomcat 6,when container is writing translated servlet then it is placing only \_jspService() and \_jspInit() method inside the translated servlet.so you can override jspInit() and you can not override \_jspInit() methods.

**Q-11: What is the diff b/w jspDestroy() and \_ jspDestroy() in jsp.:**

**A:** in Tomcat 5,when container is writing translated servlet then it is placing only \_jspService() method inside the translated servlet.so you can override both jspDestroy() and \_jspDestroy() methods. in Tomcat 6,when container is writing translated servlet then it is placing only \_jspService() and \_jspInit() method inside the translated servlet.so you can override jspDestroy() and you can not override \_jspDestroy() methods.

**Q-12: How many times \_jspInit() method will be called ?**

**A:** only once

**Q-13: How many times \_jspService() method will be called ?**

**A:** every time when client will send the request.

**Q-14: How many times \_jspDestroy() method will be called ?**

**A:** only once

**Q-15: Which web component is Faster – either Servlet or JSP.**

**A: first Request:** Servlet is faster then JSP when you send the first request

**Second Request Onwards:** Both Servlet and JSP takes the same amount of time processing 2nd Request Onwards

**Q-16: What is implicit object. how many type of implicit object. Explain with**

**the type also.**

**A:** JSP implicit Objects are the objects which are readly available in every jsp.you can not define these objects your own in JSP page.

Implicit Object is an object which can be used by you directly without creating or getting the object. there are 9 Implicit Objects in the JSP.

**JSP Implicit Object Type JSP Implicit Object**

HttpServletRequest request

HttpServletResponse response

PageContext pageContext

HttpSession session

ServletContext application

ServletConfig config

JspWriter out

Object page

Throwable exception

**Q-17: Which All implicit object .**

**A:**

## 1-The request Object:

The request object is an instance of a javax.servlet.http.HttpServletRequest object. Each time a client requests a page the JSP engine creates a new object to represent that request.

The request object provides methods to get HTTP header information including form data, cookies, HTTP methods etc.

## 2-The response Object:

The response object is an instance of a javax.servlet.http.HttpServletResponse object. it also creates an object to represent the response to the client.

The response object also defines the interfaces that deal with creating new HTTP headers. Through this object the JSP programmer can add new cookies or date stamps, HTTP status codes etc.

## 3-The out Object:

The out implicit object is an instance of a javax.servlet.jsp.JspWriter object and is used to  output content to be sent in client response.

## 4-The session Object:

The session object is an instance of javax.servlet.http.HttpSession and behaves exactly the same way that session objects behave under Java Servlets.

The session object is used to track client session between client requests.

### 5-application Object

JSP application implicit object is instance of javax.servlet.ServletContext  implementation and it’s used to get the context information and attributes in JSP. It gives facility for a JSP page to obtain and set information about the web application in which it is running.

### 6-config Object

JSP config implicit object is instance of javax.servlet.ServletConfig implementation and used to get the JSP init params ( initialization parameters ) configured in deployment descriptor.

## 7-The pageContext Object:

The pageContext object is an instance of a javax.servlet.jsp.PageContext object. The pageContext object is used to represent the entire JSP page. We can use pageContext to get and set attributes with different scopes and to forward request to other resources.

### 8-page Object

JSP page implicit object is instance of java.lang.Object class and represents the current JSP page. page object provide reference to the generated servlet class.

### 9-exception Object

JSP exception implicit object is instance of java.lang.Throwable class and used to provide exception details in JSP error pages

**Q-18: What is the difference between Page and PageContext?**

**A:** **1-** JSP page implicit object is instance of java.lang.Object class and represents the current JSP page. page object provide reference to the generated servlet class. It can be thought of as an object that represents the entire JSP page.  
**2-** The pageContext object is used to represent the entire JSP page. pageContext to get and set attributes with different scopes and to forward request to other resources.

**3-** Page context also has methods for including another URL's contents, and for forwarding or redirecting to another URL

**Q-19: What is the difference between ServletContext and PageContext?**

**A:** **ServletContext:** gives information about the container in which the servlet (or JSP) is running in.

**PageContext:**  gives the servlet (or JSP) information about the request that it is currently handling and contains information about the request and any parameters, the session, the response object

**Q-20: What are the Scripting elements in JSP.explain each completely.**

**A:** The scripting elements provides the ability to insert java code inside the jsp. There are three types of scripting elements:

1. scriptlet tag
2. expression tag
3. declaration tag

**Q-21: Explain the following 1->Scriptlet**

**2->Expression**

**3->Decelaration**

**A:**

**1-Scriptlets tag->**

(1)A scriptlet tag is used to execute java source code in JSP.

(2)Any valid java Statements allowed inside the Scriptlets.

(3)All the Statements inside the Scriptlet will be placed inside the \_jspService() method of translated servlet.

Syntax: <% java source code %>

**2-Expressions tag->**

(1)Expressions is short-cut of out.print(exp).

(2)All the Expression will be converted to out.print() and will be placed

inside the \_jspService() method of translated servlet.

(3)Expression should not end with ;(Semicolon)

Syntex : <%= statement %>

**3-Declarations tag->**

(1)Method definitions,Blocks,Constructor,class level varables are allowed inside the Decelaration tag.

(2)The code written inside the jsp declaration tag is placed outside the service() method of auto generated servlet. So it doesn't get memory at each request.

Syntex: <%! statement %>

**Q-22: What are the diff b/w Scriptlet and Deceleration.**

**A:** **Jsp Scriptlet Tag🡪**

1- The jsp scriptlet tag can only declare variables not methods.

2- The declaration of scriptlet tag is placed inside the \_jspService() method.

**Jsp Declaration Tag🡪**

1-The jsp declaration tag can declare variables as well as methods.

2-The declaration of jsp declaration tag is placed outside the \_jspService() method.

**Q-23: What is the diff b/w out.write() and out.println()**

**A: out.write() ->**will be applicable only String

**out.println()->**will be applicable to String as well as integer

**out.write("")** method is for java.io.Writer it is used to write the any file like text,csv.

**out.println("Hello")** this one is servlet method and it is use for write data on browser.

**String s = null;**

out.print(s); // outputs the text "null"

out.write(s); // NullPointerException

**Q-24: What are directive in JSP.**

**A:** (1)Directive is a special instruction to container to perform the required task.

(2)there are 3 Directives in JSP

1- include directives

2- page directives

3- taglib directives

(3)Syntax of JSP Directive

<%@ directive attribute="value" %>

**Q-25: Expalin the following 1->include directive**

**2->page directive**

**3->taglib directive**

**1- include directives🡪**

(1)include directives is used to include one jsp or html in another JSP.

(2)include directives line will be replaced with the content of include JSP or HTML at translation time.

(3)you can write one or more include directives inside the JSP.

(4)syntax:

<%@ include file="some jsp or html" %>

ex->

<%@ include file="header.html" %>

<%@ include file="login.jsp" %>

<%@ include file="footer.html" %>

**2- page directives🡪**

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

Syntax:

<%@page attribute="value" %>

**Q-26: What are various attributes Of page directive?**

**A:** Attributes of JSP page directive->

<%@page

language

import

session

extends

isThreadSafe

errorPage

isErrorPage

isELIgnored

autoFlush

%>

**1-language->**

(1)language attribute is used to specify the language for the scriptlets and declarations

(2)currently valid value is java and is the default value.

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

<%@page language="c" %> invalid(invalid language attribute)

**2-import->**

(1)import attribute is used to specify the packages to be imported for the translated servlet

(2)you can specify one or more packages with comma(,)seperation using this attribute.

<%@page import="java.util.\*","java.io.\*" %>

**3-session->**

(1) session attribute is used to enable or disable session object in the JSP.

(2) by default session object is enabled in jsp.

(3) if you want to disable use the following:

<%@page session="false" %>

**4-extends->**

(1)HttpJspBase is the default super class for the translated servlet

(2)if you want to use other super class for the translated servlet instead of HttpJspBase,use extends attribute.

(3)when you take HttpServlet as a super class then you have to override any of methods of HttpServlet.

<%@page extends="javax.servlet.http.HttpServlet" %>

**5-isThreadSafe->**

(1) This attribute is used to specify the ServletThread model required.

(2) Default Servlet Thread model is multi-thread model.

(3) Default value of isThreadSafe attribute is true.

(4) if you want to follow Single thread model,use this attribute value as false

<%@page isThreadSafe="false"%>

(5) if you want to follow Single thread model,use this attribute value as false

<%@page isThreadSafe="true"%>

|  |  |
| --- | --- |
| **1.index.jsp (generate exception)** | **2.error.jsp (Handle exception)** |
| <%@page errorPage="error.jsp"%>  <html><body><h1>  this is jsp<br/>  <%  int x=10/0;  %>  </h1></body></html> | <%@page isErrorPage="true"%>  <html><body>  <font color="red" size="6">ERROR OCCURED</br>  <%=exception%>  </font>  </body></html> |

**6-errorPage and isErrorPage->**

(1)if want to handle the exception in a centralized JSP then use these two attributes.

(2)error page is used to specify the centeralized JSP

(3)isErrorPage is used to enable exception implicit object in the centeralized JSP

used for error handling

**7-isELIgnored->**

isELIgnored attribute is used to enable or disable EL(Expression Language)

**3- taglib directives->**

taglib directives is used to use the custom tags in jsp.

**Q-27: How to disable session in JSP?**

**A:** <%@ page session="false" %>

**Q-28: How many types of JSP Standard Actions . Explain the following JSP Standard Action 1- <jsp:include>**

**2- <jsp:forward>**

**3- <jsp:useBean>**

**4- <jsp:setProperty>**

**5- <jsp:getProperty>**

**6- <jsp:param>**

**A:** **JSP Actions tags->**

(1)There are many JSP action tags or elements. Each JSP action tag is used to perform some specific tasks.

(2)The action tags are used to control the flow between pages and to use Java Bean.

**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:include>:**

(1) <jsp:include> is used to include one jsp or html inside another jsp.

(2) <jsp:include> will include the response of the included page at run time

(3) syntax:

<jsp:include page="jsp or html file"/>

Ex:

<jsp:include page="header.jsp"/>

<jsp:include page="footer.html"/>

(4) <jsp:include> functionality is similar to include() method of RequestDispathcher.

**<jsp:forward>:**

(1) <jsp:forward> is used to forward the control from one jsp to another jsp or html

(2) syntax:

<jsp:forward page="jsp or html file"/>

Ex:

<jsp:forward page="header.jsp"/>

(3) <jsp:forward> functionality is similar to forward() method of RequestDispathcher.

**<jsp:param>:**

(1) <jsp:param> is used to send the parameter data from one jsp to another jsp

while you are including or forwarding.

(2) <jsp:param> must be used along with <jsp:include> or <jsp:forward>

(3) Syntax:

<jsp:param name="email" value="subhag@gmail.com"/>

Ex:

<jsp:include page="header.jsp"/>

<jsp:param name="email" value="subhag@gmail.com"/>

</jsp:include>

<jsp:include page="home.jsp"/>

<jsp:param name="email" value="nidhi@gmail.com"/>

</jsp:forward>

**<jsp:useBean>,<jsp:setProperty>,<jsp:getProperty>**

(1)theses 3 tags are used for

- creating java Bean object

- storing data in java Bean Object

- Accessing data from java Bean object

**<jsp:useBean>->**

<jsp:useBean> is used to create the java Bean object and to store that object in the required scope.

syntax:

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

ex:

<jsp:useBean id="ST" class="Student" scope="session"/>

**<jsp:setProperty>**

<jsp:setProperty> is used to set the value to the property of java bean object.

syntax:

<jsp:setProperty name="id/attName" property="<name\_of\_variable>" values="..."/>

ex:

<jsp:setProperty name="ST" property="name" values="sri"/>

**<jsp:getProperty>**

<jsp:getProperty> is used to s=get the value to the property of java bean object.

syntax:

<jsp:getProperty name="id/attName" property="<name\_of\_variable>" />

ex:

<jsp:getProperty name="ST" property="name" />

**Q-29: What are the options in JSP to include files?**

**A:** In JSP, we can perform inclusion in the following ways:

1->**By include directive:** For example:

<%@ include file=”header.jsp” %>

2->**By include action:** For example:

<jsp:include page="header.jsp">

3->**By using pageContext implicit object** For example:

<%

pageContext.include(“/header.jsp”);

%>

4->**By using RequestDispatcher object:** For example:

<%

RequestDispatcher rd = request.getRequestDispatcher(“/header.jsp”);

Rd.incliude(request,response);

%>

**Q-30: What is difference between include directive and include action?**

**A :**

|  |  |
| --- | --- |
| **JSP include directive** | **JSP include action** |
| 1-includes resource at translation time. | includes resource at request time. |
| 2-Better for static pages. | Better for dynamic pages. |
| 3-includes the original content in the Generated servlet. | calls the include method. |
| 4-Attribute of include directive is file  <%@include file="header.jsp"@%> | Attribute of include directive is page  <jsp:include page="header.jsp"> |
| 5- include directive line will be replaced  content of included jsp or html  translation time. | include action line will be replaced with  with response of jsp or html included at  at run time. |
| 6- include directive is not good in  performance. | include action is good in the performance. |
| 7-when you are including any jsp using  directive,you can not pass parameters | when you are including any jsp using include action ,you can pass parameters to jsp using<jsp:param> |

**Q-31: Diff between <jsp:include page=…> and <%@include file=…>.**

**A:** Both these tags include information from 1 page to another.

1->The first tag acts as a function call between two Jsp’s. It is executed each time client page is accessed by the client. It is useful to modularize the web application. New content is included in the output.

2->The second tag content of file is textually embedded having similar directive. The changed content is not included in the output. It is helpful when code from one jsp is required by several jsp’s.

**Q-32: How do you pass control from one JSP page to another?**

**A:** Use the following ways to pass control of a request from one servlet to another or one jsp to another:

1->The RequestDispatcher object ‘s forward method to pass the control.

2->Using the response.sendRedirect method.

**Q-33: What are the three kind of comment in jsp?**

**A:** These are the three types of commenst in jsp:

1->JSP Comment: <%-- this is jsp comment -- %>

2->HTML Comment: <!-- this is HTMl comment -- >

3->Java Comments: <% // single line java comment /\* this is multiline comment \*/ %>

**Q-34: What is JSTL? What the different types of JSTL tags are ? explain the following JSTL tags**

1. **Core Tag**
2. **Formating Tags**
3. **SQL Tags**
4. **XML tags**

**A: A:** (1)jstl stands for jsp standard tag library

(2)jstl is collection of utility tags implemented by SUN which can be used by developer in jsp.

(3)jstl tags are divided into following types.

--> Core Tags

--> Formating Tags

--> SQL Tags

--> XML Tags

(4)you have to do the following to use JSTL Core tags:

(A)Copy jstl1-2.jar to your project WEB-INF/lib folder-- (TOMCAT\_HOME\webapps\examples\WEB-INF/lib)

**(B)Use the following taglib directive in any jsp**

**1-Core Tags:->**Core tags provide support for iteration, conditional logic, catch exception, url, forward or redirect response etc. To use JSTL core tags,.

we should include it in the JSP page like below.

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

**2-Formatting and Localization Tags**:-> These tags are provided for formatting of Numbers, Dates and i18n support through locales and resource bundles. We can include these tags in JSP with below syntax

<%@taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="fmt"%>

**3-SQL Tags:->** JSTL SQL Tags provide support for interaction with relational databases such as Oracle, MySql etc. Using SQL tags we can run database queries, we include it in JSP with below syntax:

<% @taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>

**Q-35: Explain all JSTL Core tags**

1. **<c:out> (2) <c:set> (3) <c:remove> (4) <c:if> (5) <c:choose> (6)<c:when>**

**(7) <c:otherwise> (8) <c:forEach> (9) <c:import> (10) <c:redirect>**

**(11) <c:param>**

**(1)<c:out>->**

<c:out> is used to display data to jsp output stream

attribute : value

Usage :

<c:out value="sri"/>

<c:out value="msg"/>

<c:out value="${msg}"/>

**(2)<c:set>->**

<c:set> is used to set the attribute in the required scope.

attribute : var,value,scope

Usage :

<c:set var="EM" value="subhag@gmail.com" scope="session"/>

->session.setAttribute("EM","subhag@gmail.com");

**(3)<c:remove>->**

<c:remove> is used to remove the attribute from the required scope.

attribute : var,scope

Usage : <c:remove var="EM" scope="session"/>

->session.removeAttribute("EM");

**(4)<c:if>->**

<c:if> is used to perform simple conditional checks.

attribute : test

Usage :

<c:if test="${sname eq'sri'}">

your name is Sri

</c:if>

<c:if test="${sname ne'sri'}">

your name is Not Sri

</c:if>

**(5)<c:choose>,(6)<c:when>,(7)<c:otherwise->**

these tags are used to perform simple conditional checks like switch statement

attribute : -- choose -> no attribute available

-- when -> test

-- otherwise -> no attribute available

Usage :

<c:choose>

<c:when test="${sid eq 99}">

ID is 99

</c:when>

<c:when test="${sid eq 88}">

ID is 88

</c:when>

<c:otherwise>

ID is unknown

</c:otherwise>

</c:choose>

**(8)<c:forEach>->**

<c:forEach> is used to access the element of collection ar array

attribute : var,items,start,end,step,varStatus

Usage :

using <c:forEach>,you can access the following type of data

1- Collection of Strings,Wrappers and Date

2- Collection of Collections

3- Collection of User defined class object

4- Map objects

5-Collection of Map objects

**(9)<c:import>->**

<c:import> is used to include the resource(html or jsp)

attribute : url

usgae :

<jsp:include page="header.jsp"/>

<c:import url="header.jsp"/>

<c:import url="http://www.google.co.in"/>

<c:import url="http://www.kclink.com/services.jsp"/>

note-><jsp:include> can include the resources which are in the same server where as <c:import> can include the resources which are in the same server and different server

**(10)<c:redirect>**

<c:redirect> is used to redirect the request to the specified resource(html or jsp)

attribute : url

Usage :

<c:redirect url="header.jsp"/>

<c:redirect url="http://www.google.co.in"/>

<c:redirect url="http://www.kclink.com/services.jsp"/>

note-><jsp:forward> can forward request to resources which are in the same application where as <c:redirect> can forward request to the resources which are in same application and different application running in the same server or different server

**(11)<c:param>->**

<c:param> is used to define parameters

<c:param> must be used along with <c:import> and <c:redirect>

attributes : name,value

Usage :

<c:param name = "email" value="subhag@gmail.com">

Ex:

1-<c:import url="footer.jsp">

<c:param name="company name" value="KC Link Pvt Ltd.">

</C:import>

2-<c:redirect url="home.jsp">

<c:param name="company name" value="KC Link Pvt Ltd.">

</C:redirect>

**(12)<c:url>**

<c:url> is used to encode the url with sessionid

attributes : value

usage :

<c:url value="hello.jsp">

<a href="hello.jsp">Click Here</a>

<a href='<c:url value=hello.jsp"/>'>Click Here</a>

<a href='<%=response.encodeURL("hello.jsp")%>'>Click Here</a>

**Q-36: What is the diff b/w <jsp:include> and <c:import>**

A: <jsp:include> can include the resources which are in the same server where as <c:import> can include the resources which are in the same server and different server

**Q-37: What is the diff b/w <jsp:forward> and <c:redirect>**

A: <jsp:forward> can forward request to resources which are in the same application where as <c:redirect> can forward request to the resources which are in same application and different application running in the same server or different server

**Q-37: what is the use of <c:out> .even we have EL expression**

**Q-38: What is a JSP Custom Tags?**

**A:**

(1)when you want to display the dynamic content in the JSP then you can use the EL Expression and JSTL Tags.

(2)if these two are not enough for your application requirement then you can develop your own tags called Custom Tags.

(3)when you develop a custom tag then that can be reused across multiple JSP,s.

**Q-39: What are the steps to create Custom Tags.**

**A:** **Steps to develop Custom tags->**

1-identify the information related to the tag which you want to develop includes.

--> name of the tag

--> body-content

--> attributes allowed for the tag

etc..

2-write the tag handler class with the following steps:

--> define java class by extending TagSupport or BodyTagSupport class.

--> define the variables for the attributes defined for the tag.

--> define the setter method for the variables

--> override the required lifecycle methods.

3-place one TLd file in WEB-INF or subdirectory

4-specify the tag information in the tid file which includes

--> URI

--> name of the tag

--> body-content

--> tag-class

--> attributes allowed for the tag

**Steps to use Custom tags->**

1-use the taglib directive to use custom tags in jsp

2-taglib directive has two attributed called prefix and uri

3-URI of the taglib directive must match with URI specified in TLD file

4-Prefix of the taglib directive can be anything as you like.

5->refer the tag in the jsp using prefix.

Ex:

<kclink:showMessage sname='${param.studname}' email='{studemail}'/>

<kclink:showMessage sname='Sri' email='sri@gmail.com'/>

**Q-40: What is El expression. What is the need of EL expression.**

**A:** JSP Expression Language (EL) makes it possible to easily access application data stored in JavaBeans components Without using the java code.

### Syntax : ${ expression }

**Q-41: How many EL implicit object.**

**A:** following are various EL implicit objects.

(1)param (2)param Values (3)header (4)headerValues

(5)cookie (6)pageScope (7)requestScope (8)sessionScope

(9)applicationScope (10)initParam (11)pageContext(\*)

**param and paramValues->**

(1)these two EL implicit objects are used to collect the request parameters.

(2)when use param then internally requset.getParameter() method will be called.

(3)when use param Values then internally requset.getParameter() method will be called.

Usage:

- ${param.email}

- ${param.course}

- ${paramValues.course[0]}

- ${paramValues.course[1]}

**header and headerValues->**

(1)these two EL implicit objects are used to collect the request Headers.

(2)when use header then internally requset.getHeader() method will be called.

(3)when use headerbValues then internally requset.getHeaderValues() method will be called.

Usage:

- ${header.host}

- ${header.refer}

- ${headerValues.accept-language[0]}

- ${headerValues.accept-language[1]}

**cookie->**

this EL implicit object is used to collect the request Cookies.

Usage:

-${cookie.<cookieName>.value}

ex: ${cookie.email.value}

**initParam->**

this EL implicit object is used to collect the context parameter from ServletContext object

Usage:

-${initParam.state}

-${initParam.city}

**pageContext->**

this EL implicit object allows to access all other jsp implicit objects and their properties.

Usage:

-${pageContext.session.id}

-${pageContext.session.id.length}

-${pageContext.request.remoteAddr}

-${pageContext.request.method}

-${pageContext.request.requestURI}

-${pageContext.response.contentType}

**Q-42: How can we disable EL ?**

**A:** We can disable using isELIgnored attribute of the page directive:

**<%@ page isELIgnored ="true|false" %>**If it is true, EL expressions are ignored when they appear in static text or tag attributes. If it is false, EL expressions are evaluated by the container. .

**Q-43: What is the different scope available in jsp?**

threre are 4 scopes available in servlet

(1) page Scope

(2) Request Scope

(3) Session Scope

(4) Context Scope

**(1) page Scope**->

the data from page scope can be accessed within the same jsp only.

(2)**Request Scope**->

(1)when the data will be stored in HttpServletRequest object and the scope will be request scope

(2)the data from request scope can be accessed by that single user in that request only befor sending the response to the client.

(3)**Session Scope**->

(1)when the data will be stored in the HttpSession object then the scope will be session scope.

(2)the data from session scope can be accessed by single user across multiple request.

(4)**Context or Application Scope**->

(1)when the data will be stored in ServletContext object then the scope will be context scope.

(2)the data from context or application scope can be accessed by multiple users across multiple requests.

**Q-44: When to use application scope?**

**A:** If we want to make our data available to the entire application then we have to use application scope.

**Q-45: What is the difference between variable declared inside the declaration tag and variable declared in scriptlet?**

**A**: Variable declared inside declaration part is treated as a instance variable and will be placed directly at class level in the generated servlet. Variable declared in a scriptlet will be placed inside \_jspService () method of generated servlet. It acts as local variable.

**Q-46: In JSP page how can we handle runtime exception?**

**A:** We can use the errorPage attribute of the page directive to have uncaught run-time exceptions automatically forwarded to an error processing page.

Example: **<%@ page errorPage="error.jsp" %>**

It will redirect the browser to the JSP page error.jsp if an uncaught exception is encountered during request processing. Within error.jsp, will have to indicate that it is an error-processing page, using the directive:

**<%@ page isErrorPage="true" %>**

**Q-47: How can I implement the thread safe jsp page?**

**A:** You can make your JSPs thread-safe by having them implement the SingleThreadModel interface. This is done by adding the directive in the JSP.

**<%@ page isThreadSafe="false" %>**

**Q-48: Which directive is used in jsp custom tag?**

**A:** The jsp taglib directive.

**Q-49: What are the 3 tags used in JSP bean development?**

**A:** 1->jsp:useBean

2->jsp:setProperty

3->jsp:getProperty

**Q-50:Can we use the exception implicit object in any jsp page ?**

**A:**No. The exception implicit object can only be used in the error page which defines it with the isErrorPage attribute of page directive.

**Q-51:if I will create a JSP page and in jsp page I will create two text field**

**usename**

**password**

submit

**if I will click on submit button then what is the flow of execution**

**A:**

**1-**Client sends the request for a resource called login.html.

**2-**Response of login.html will be displayed to Client

**3-**Client enters the username and password and submit the form

**4-**Browser collects the form data & from action and sends to server via HTTP

**5-**Server receives the request and delegates the Web Container.

**6-**Web Container collects the form action value i.e URL Pattern.

**7-**Web Container checks whether any servlet is configured with the matching URL Pattern.

**8-**Check from the XML file either Annotation.

**9-**After identifying the servlet class,Web Container performs the servlet life cycle process.

**10-**Finally response of LoginServlet will be delivered to client.