JSP - Interview Questions

Dear readers, these **JSP Interview Questions** have been designed specially to get acquainted with the nature of questions you may encounter during your interview for the subject of **JSP**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer –

What is JSP?

JavaServer Pages (JSP) is a technology for developing Webpages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

What are advantages of using JSP?

JSP offer several advantages as listed below –

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

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.

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.

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

What are the advantages of JSP over Active Server Pages (ASP)?

The advantages of JSP are twofold.

First, the dynamic part is written in Java, not Visual Basic or other MS specific language, so it is more powerful and easier to use.

Second, it is portable to other operating systems and non-Microsoft Web servers.

What are the advantages of JSP over Pure Servlets?

It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML. Other advantages are −

- Embedding of Java code in HTML pages.
- Platform independence.
- Creation of database-driven Web applications.
- Server-side programming capabilities.

What are the advantages of JSP over Server-Side Includes (SSI)?

SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.

What are the advantages of JSP over JavaScript?

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.

What are the advantages of JSP over Static HTML?

Regular HTML, of course, cannot contain dynamic information.

Explain lifecycle of a JSP.

A JSP Lifecycle consists of following steps −

Compilation − When a browser asks for a JSP, the JSP engine first checks to see whether it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps −

- Parsing the JSP.
- Turning the JSP into a servlet.
- Compiling the servlet.

Initialization – When a container loads a JSP it invokes the jspInit() method before servicing any requests

Execution – Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the _jspService() method in the JSP.The _jspService() method of a JSP is invoked once per a request and is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods ie. GET, POST, DELETE etc.

Cleanup – The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.The jspDestroy() method is the JSP equivalent of the destroy method for servlets.

What is a sciptlet in JSP and what is its syntax?

A scriptlet can contain any number of JAVA language statements, variable or method declarations, or expressions that are valid in the page scripting language.

Following is the syntax of Scriptlet –

```
<% code fragment %>
```

What are JSP declarations?

A declaration declares one or more variables or methods that you can use in Java code later in the JSP file. You must declare the variable or method before you use it in the JSP file.

```
<%! declaration; [ declaration; ]+ ... %>
```

What are JSP expressions?

A JSP expression element contains a scripting language expression that is evaluated, converted to a String, and inserted where the expression appears in the JSP file.

The expression element can contain any expression that is valid according to the Java Language Specification but you cannot use a semicolon to end an expression.

Its syntax is –

```
<%= expression %>
```

What are JSP comments?

JSP comment marks text or statements that the JSP container should ignore. A JSP comment is useful when you want to hide or "comment out" part of your JSP page.

Following is the syntax of JSP comments –

```
<%-- This is JSP comment --%>
```

What are JSP Directives?

A JSP directive affects the overall structure of the servlet class. It usually has the following form –

```
<%@ directive attribute = "value" %>
```

What are the types of directive tags?

The types directive tags are as follows –

<%@ page ... %> – Defines page-dependent attributes, such as scripting language, error page, and buffering requirements.

<%@ include ... %> – Includes a file during the translation phase.

<%@ taglib ... %> – Declares a tag library, containing custom actions, used in the page.

What are JSP actions?

JSP actions use constructs in XML syntax to control the behavior of the servlet engine. You can dynamically insert a file, reuse JavaBeans components, forward the user to another page, or generate HTML for the Java plugin.

Its syntax is as follows –

```
<jsp:action_name attribute = "value" />
```

Name some JSP actions.

jsp:include, jsp:useBean,jsp:setProperty,jsp:getProperty, jsp:forward,jsp:plugin,jsp:element, jsp:attribute, jsp:body, jsp:text

What are JSP literals?

Literals are the values, such as a number or a text string, that are written literally as part of a program code. The JSP expression language defines the following literals –

Boolean – true and false

Integer – as in Java

Floating point – as in Java

String – with single and double quotes; " is escaped as \", ' is escaped as \', and \ is escaped as \\.

Null – null

What is a page directive?

The **page** directive is used to provide instructions to the container that pertain to the current JSP page. You may code page directives anywhere in your JSP page.

What are various attributes Of page directive?

Page directive contains the following 13 attributes.

1. language
2. extends
3. import
4. session
5. isThreadSafe
6. info
7. errorPage
8. isErrorpage
9. contentType
10. isELIgnored
11. buffer
12. autoFlush
13. isScriptingEnabled

What is a buffer attribute?

The buffer attribute specifies buffering characteristics for the server output response object.

What happens when buffer is set to a value *"none"* ?

When buffer is set to *"none"*, servlet output is immediately directed to the response output object.

What is autoFlush attribute?

The **autoFlush** attribute specifies whether buffered output should be flushed automatically when the buffer is filled, or whether an exception should be raised to indicate buffer overflow.

A value of **true** (default) indicates automatic buffer flushing and a value of **false** throws an exception.

What is contentType attribute?

The **contentType** attribute sets the character encoding for the JSP page and for the generated response page. The default content type is text/html, which is the standard content type for HTML pages.

What is errorPage attribute?

The **errorPage** attribute tells the JSP engine which page to display if there is an error while the current page runs. The value of the errorPage attribute is a relative URL.

What is *isErrorPage* attribute?

The *isErrorPage* attribute indicates that the current JSP can be used as the error page for another JSP.

The value of *isErrorPage* is either true or false. The default value of the *isErrorPage* attribute is false.

What is extends attribute?

The **extends** attribute specifies a superclass that the generated servlet must extend.

What is import attribute?

The **import** attribute serves the same function as, and behaves like, the Java import statement. The value for the import option is the name of the package you want to import.

What is info attribute?

The **info** attribute lets you provide a description of the JSP.

What is isThreadSafe attribute?

The **isThreadSafe** option marks a page as being thread-safe. By default, all JSPs are considered thread-safe. If you set the isThreadSafe option to false, the JSP engine makes sure that only one thread at a time is executing your JSP.

What is language attribute?

The **language** attribute indicates the programming language used in scripting the JSP page.

What is session attribute?

The **session** attribute indicates whether or not the JSP page uses HTTP sessions. A value of true means that the JSP page has access to a builtin session object and a value of false means that the JSP page cannot access the builtin session object.

What is isELIgnored Attribute?

The **isELIgnored** option gives you the ability to disable the evaluation of Expression Language (EL) expressions.

The default value of the attribute is true, meaning that expressions, ${...}, are evaluated as dictated by the JSP specification. If the attribute is set to false, then expressions are not evaluated but rather treated as static text.

What is isScriptingEnabled Attribute?

The **isScriptingEnabled** attribute determines if scripting elements are allowed for use.

The default value (true) enables scriptlets, expressions, and declarations. If the attribute's value is set to false, a translation-time error will be raised if the JSP uses any scriptlets, expressions (non-EL), or declarations.

What is a include directive?

The include directive is used to includes a file during the translation phase. This directive tells the container to merge the content of other external files with the current JSP during the translation phase. You may code include directives anywhere in your JSP page.

The general usage form of this directive is as follows −

```
<%@ include file = "relative url" >
```

What is a taglib directive?

The taglib directive follows the following syntax −

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

uri attribute value resolves to a location the container understands

prefix attribute informs a container what bits of markup are custom actions.

The taglib directive follows the following syntax –

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

What do the **id** and **scope** attribute mean in the action elements?

Id attribute – The id attribute uniquely identifies the Action element, and allows the action to be referenced inside the JSP page. If the Action creates an instance of an object the id value can be used to reference it through the implicit object PageContext

Scope attribute – This attribute identifies the lifecycle of the Action element. The id attribute and the scope attribute are directly related, as the scope attribute determines the lifespan of the object associated with the id. The scope attribute has four possible values: (a) page, (b)request, (c)session, and (d) application.

What is the function of <jsp:include> action?

This action lets you insert files into the page being generated. The syntax looks like this –

```
<jsp:include page = "relative URL" flush = "true" />
```

Where **page** is the relative URL of the page to be included.

Flush is the boolean attribute the determines whether the included resource has its buffer flushed before it is included.

What is the difference between include action and include directive?

Unlike the **include directive**, which inserts the file at the time the JSP page is translated into a servlet, **include action** inserts the file at the time the page is requested.

What is <jsp:useBean> action?

The **useBean** action is quite versatile. It first searches for an existing object utilizing the id and scope variables. If an object is not found, it then tries to create the specified object.

The simplest way to load a bean is as follows –

```
<jsp:useBean id = "name" class = "package.class" />
```

What is <jsp:setProperty> action?

The **setProperty** action sets the properties of a Bean. The Bean must have been previously defined before this action.

What is <jsp:getProperty> action?

The **getProperty** action is used to retrieve the value of a given property and converts it to a string, and finally inserts it into the output.

What is <jsp:forward> Action?

The **forward** action terminates the action of the current page and forwards the request to another resource such as a static page, another JSP page, or a Java Servlet.

The simple syntax of this action is as follows −

```
<jsp:forward page = "Relative URL" />
```

What is <jsp:plugin> Action?

The **plugin** action is used to insert Java components into a JSP page. It determines the type of browser and inserts the <object> or <embed> tags as needed.

If the needed plugin is not present, it downloads the plugin and then executes the Java component. The Java component can be either an Applet or a JavaBean.

What are the different scope values for the JSP action?

The scope attribute identifies the lifecycle of the Action element. It has four possible values: (a) page, (b)request, (c)session, and (d) application.

What are JSP implicit objects?

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

What implicit objects are supported by JSP?

request, response, out, session, application, config, pageContext, page, Exception

What is a 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.

How can you read a request header information?

Using getHeaderNames() method of HttpServletRequest to read the HTTP header information. This method returns an Enumeration that contains the header information associated with the current HTTP request.

What is a response object?

The response object is an instance of a javax.servlet.http.HttpServletRequest object. Just as the server creates the request 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.

What is the **out** implicit object?

The **out** implicit object is an instance of a javax.servlet.jsp.JspWriter object and is used to send content in a response.

What is the difference between JspWriter and PrintWriter?

The **JspWriter** object contains most of the same methods as the java.io.PrintWriter class. However, JspWriter has some additional methods designed to deal with buffering. Unlike the **PrintWriter** object, JspWriter throws IOExceptions.

What is the session Object?

The session object is an instance of javax.servlet.http.HttpSession and is used to track client session between client requests

What is an application Object?

The application object is direct wrapper around the ServletContext object for the generated Servlet and in reality an instance of a javax.servlet.ServletContext object.

This object is a representation of the JSP page through its entire lifecycle. This object is created when the JSP page is initialized and will be removed when the JSP page is removed by the jspDestroy() method.

What is a config Object?

The config object is an instantiation of javax.servlet.ServletConfig and is a direct wrapper around the ServletConfig object for the generated servlet.

This object allows the JSP programmer access to the Servlet or JSP engine initialization parameters such as the paths or file locations etc.

What is a 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.

This object stores references to the request and response objects for each request. The application, config, session, and out objects are derived by accessing attributes of this object.

The pageContext object also contains information about the directives issued to the JSP page, including the buffering information, the errorPageURL, and page scope.

What is a page object?

This object is an actual reference to the instance of the page. It can be thought of as an object that represents the entire JSP page.

The page object is really a direct synonym for the this object.

What is an exception Object?

The exception object is a wrapper containing the exception thrown from the previous page. It is typically used to generate an appropriate response to the error condition.

What is difference between GET and POST method in HTTP protocol?

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ? Character.

The POST method packages the information in exactly the same way as GET methods, but instead of sending it as a text string after a ? in the URL it sends it as a separate message. This message comes to the backend program in the form of the standard input which you can parse and use for your processing.

How to read form data using JSP?

JSP handles form data parsing automatically using the following methods depending on the situation −

getParameter() − You call request.getParameter() method to get the value of a form parameter.

getParameterValues() − Call this method if the parameter appears more than once and returns multiple values, for example checkbox.

getParameterNames() − Call this method if you want a complete list of all parameters in the current request.

getInputStream() − Call this method to read binary data stream coming from the client.

What are filters?

JSP Filters are Java classes that can be used in JSP Programming for the following purposes −

To intercept requests from a client before they access a resource at back end.

To manipulate responses from server before they are sent back to the client.

How do you define filters?

Filters are defined in the deployment descriptor file web.xml and then mapped to either servlet or JSP names or URL patterns in your application's deployment descriptor.

When the JSP container starts up your web application, it creates an instance of each filter that you have declared in the deployment descriptor. The filters execute in the order that they are declared in the deployment descriptor.

What are cookies?

Cookies are text files stored on the client computer and they are kept for various information tracking purpose.

How cookies work?

Cookies are usually set in an HTTP header (although JavaScript can also set a cookie directly on a browser).If the browser is configured to store cookies, it will then keep this information until the expiry date. If the user points the browser at any page that matches the path and domain of the cookie, it will resend the cookie to the server.

How do you set cookies in the JSP?

Setting cookies with JSP involves three steps –

Creating a Cookie object – You call the Cookie constructor with a cookie name and a cookie value, both of which are strings.

Setting the maximum age – You use setMaxAge to specify how long (in seconds) the cookie should be valid.

Sending the Cookie into the HTTP response headers – You use response.addCookie to add cookies in the HTTP response header

How to read cookies with JSP?

To read cookies, you need to create an array of javax.servlet.http.Cookie objects by calling the getCookies( ) method of HttpServletRequest. Then cycle through the array, and use getName() and getValue() methods to access each cookie and associated value.

How to delete cookies with JSP?

To delete cookies is very simple. If you want to delete a cookie then you simply need to follow up following three steps –

Read an already existing cookie and store it in Cookie object.

Set cookie age as zero using **setMaxAge()** method to delete an existing cookie.

Add this cookie back into response header.

How is Session Management done in JSP?

Session management can be achieved by the use of –

Cookies – A webserver can assign a unique session ID as a cookie to each web client and for subsequent requests from the client they can be recognized using the received cookie.

Hidden Form Fields – A web server can send a hidden HTML form field along with a unique session ID as follows –

```
<input type = "hidden" name = "sessionid" value = "12345">
```

This implies that when the form is submitted, the specified name and value will be getting included in GET or POST method.

URL Rewriting – In URL rewriting some extra information is added on the end of each URL that identifies the session. This URL rewriting can be useful where a cookie is disabled.

The session Object – JSP makes use of servlet provided HttpSession Interface which provides a way to identify a user across more than one page request or visit to a Web site and to store information about that user.

How can you delete a session data?

When you are done with a user's session data, you have several options –

Remove a particular attribute – You can call *public void removeAttribute(String name)* method to delete the value associated with the a particular key.

Delete the whole session – You can call *public void invalidate()* method to discard an entire session.

Setting Session timeout – You can call *public void setMaxInactiveInterval(int interval)* method to set the timeout for a session individually.

Log the user out – The servers that support servlets 2.4, you can call **logout** to log the client out of the Web server and invalidate all sessions belonging to all the users.

web.xml Configuration – If you are using Tomcat, apart from the above mentioned methods, you can configure session time out in web.xml file as follows.

How can you upload a file using JSP?

To upload a single file you should use a single <input .../> tag with attribute type = "file".To allow multiple files uploading, include more than one input tags with different values for the name attribute.

Where will be the uploaded files stored?

You can hard code this in your program or this directory name could also be added using an external configuration such as a context-param element in web.xml.

What is JSP page redirection?

Page redirection is generally used when a document moves to a new location and we need to send the client to this new location or may be because of load balancing, or for simple randomization.

What is the difference between <jsp:forward page = ... > and response.sendRedirect(url)?

The <jsp:forward> element forwards the request object containing the client request information from one JSP file to another file. The target file can be an HTML file, another JSP file, or a servlet, as long as it is in the same application context as the forwarding JSP file.

sendRedirect sends HTTP temporary redirect response to the browser, and browser creates a new request to go the redirected page.

What is a hit count for a Webpage?

A hit counter tells you about the number of visits on a particular page of your web site.

How do you implement hit counter in JSP?

To implement a hit counter you can make use of Application Implicit object and associated methods getAttribute() and setAttribute().

This object is a representation of the JSP page through its entire lifecycle. This object is created when the JSP page is initialized and will be removed when the JSP page is removed by the jspDestroy() method.

How can you implement hit counter to avoid loss of count data with each restart of the application?

You can follow the below steps –

Define a database table with a single count, let us say hitcount. Assign a zero value to it.

With every hit, read the table to get the value of hitcount.

Increase the value of hitcount by one and update the table with new value.

Display new value of hitcount as total page hit counts.

If you want to count hits for all the pages, implement above logic for all the pages.

What is auto refresh feature?

Consider a webpage which is displaying live game score or stock market status or currency exchange ration. For all such type of pages, you would need to refresh your Webpage regularly using refresh or reload button with your browser.

JSP makes this job easy by providing you a mechanism where you can make a webpage in such a way that it would refresh automatically after a given interval.

How do you implement the auto refresh in JSP?

The simplest way of refreshing a Webpage is using method setIntHeader() of response object. Following is the signature of this method −

```
public void setIntHeader(String header, int headerValue)
```

This method sends back header "Refresh" to the browser along with an integer value which indicates time interval in seconds.

What is JSTL?

The JavaServer Pages Standard Tag Library (JSTL) is a collection of useful JSP tags which encapsulates core functionality common to many JSP applications.

JSTL has support for common, structural tasks such as iteration and conditionals, tags for manipulating XML documents, internationalization tags, and SQL tags. It also provides a framework for integrating existing custom tags with JSTL tags.

What the different types of JSTL tags are ?

Types of JSTL tags are −

Core Tags

Formatting tags

SQL tags

XML tags

JSTL Functions

What is the use of <c:set > tag?

The <c:set > tag is JSTL-friendly version of the setProperty action. The tag is helpful because it evaluates an expression and uses the results to set a value of a JavaBean or a java.util.Map object.

What is the use of <c:remove > tag?

The <c:remove > tag removes a variable from either a specified scope or the first scope where the variable is found (if no scope is specified).

What is the use of <c:catch> tag?

The <c:catch> tag catches any Throwable that occurs in its body and optionally exposes it. Simply it is used for error handling and to deal more gracefully with the problem.

What is the use of <c:if> tag?

The <c:if> tag evaluates an expression and displays its body content only if the expression evaluates to true.

What is the use of <c:choose> tag?

The <c:choose> works like a Java switch statement in that it lets you choose between a number of alternatives. Where the switch statement has case statements, the <c:choose> tag has <c:when> tags. A a switch statement has default clause to specify a default action and similar way <c:choose> has <otherwise> as default clause.

What is the use of <c:forEach >, <c:forTokens> tag?

The <c:forEach >, <c:forTokens> tags exist as a good alternative to embedding a Java for, while, or do-while loop via a scriptlet.

What is the use of <c:param> tag?

The <c:param> tag allows proper URL request parameter to be specified with URL and it does any necessary URL encoding required.

What is the use of <c:redirect > tag?

The <c:redirect > tag redirects the browser to an alternate URL by providing automatically URL rewriting, it supports context-relative URLs, and it supports the <c:param> tag.

What is the use of <c:url> tag?

The <c:url> tag formats a URL into a string and stores it into a variable. This tag automatically performs URL rewriting when necessary.

What are JSTL formatting tags ?

The JSTL formatting tags are used to format and display text, the date, the time, and numbers for internationalized Web sites. Following is the syntax to include Formatting library in your JSP –

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

What are JSTL SQL tags?

The JSTL SQL tag library provides tags for interacting with relational databases (RDBMSs) such as Oracle, mySQL, or Microsoft SQL Server.

Following is the syntax to include JSTL SQL library in your JSP −

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

What are JSTL XML tags?

The JSTL XML tags provide a JSP-centric way of creating and manipulating XML documents. Following is the syntax to include JSTL XML library in your JSP.

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

What is a JSP custom tag?

A custom tag is a user-defined JSP language element. When a JSP page containing a custom tag is translated into a servlet, the tag is converted to operations on an object called a tag handler. The Web container then invokes those operations when the JSP page's servlet is executed.

What is JSP Expression Language?

JSP Expression Language (EL) makes it possible to easily access application data stored in JavaBeans components. JSP EL allows you to create expressions both (a) arithmetic and (b) logical. A simple syntax for JSP EL is −

```
${expr}
```

Here expr specifies the expression itself.

What are the implicit EL objects in JSP ?

The JSP expression language supports the following implicit objects −

pageScope − Scoped variables from page scope

requestScope − Scoped variables from request scope

sessionScope − Scoped variables from session scope

applicationScope − Scoped variables from application scope

param − Request parameters as strings

paramValues – Request parameters as collections of strings

headerHTTP – request headers as strings

headerValues – HTTP request headers as collections of strings

initParam – Context-initialization parameters

cookie – Cookie values

pageContext – The JSP PageContext object for the current page

How can we disable EL ?

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.

What type of errors you might encounter in a JSP code?

Checked exceptions – Achecked exception is an exception that is typically a user error or a problem that cannot be foreseen by the programmer. For example, if a file is to be opened, but the file cannot be found, an exception occurs. These exceptions cannot simply be ignored at the time of compilation.

Runtime exceptions – A runtime exception is an exception that occurs that probably could have been avoided by the programmer. As opposed to checked exceptions, runtime exceptions are ignored at the time of compliation.

Errors – These are not exceptions at all, but problems that arise beyond the control of the user or the programmer. Errors are typically ignored in your code because you can rarely do anything about an error. For example, if a stack overflow occurs, an error will arise. They are also ignored at the time of compilation.

In JSP page how can we handle runtime exception?

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" %>

What is Internationalization?

Internationalization means enabling a web site to provide different versions of content translated into the visitor's language or nationality.

What is Localization?

Localization means adding resources to a web site to adapt it to a particular geographical or cultural region for example Hindi translation to a web site.

What is locale?

This is a particular cultural or geographical region. It is usually referred to as a language symbol followed by a country symbol which are separated by an underscore. For example "en_US" represents english locale for US.

What is difference between <%-- comment --%> and <!-- comment -->?

<%-- comment --%> is JSP comment and is ignored by the JSP engine.

<!-- comment --> is an HTML comment and is ignored by the browser.

Is JSP technology extensible?

YES. JSP technology is extensible through the development of custom actions, or tags, which are encapsulated in tag libraries.

How do I include static files within a JSP page?

Static resources should always be included using the JSP **include directive**. This way, the inclusion is performed just once during the translation phase. Do note that you should always supply a relative URL for the file attribute. Although you can also include static resources using the action, this is not advisable as the inclusion is then performed for each and every request.

Can a JSP page process HTML FORM data?

Yes. However, unlike Servlet, you are not required to implement HTTP-protocol specific methods like doGet() or doPost() within your JSP page. You can obtain the data for the FORM input elements via the request implicit object within a scriptlet or expression.

How do you pass control from one JSP page to another?

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

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

Using the *response.sendRedirect* method.

Can you make use of a ServletOutputStream object from within a JSP page?

No. You are supposed to make use of only a JSPWriter object (given to you in the form of the implicit object out) for replying to clients.

A JSPWriter can be viewed as a buffered version of the stream object returned by response.getWriter(), although from an implementational perspective, it is not.

What is the page directive is used to prevent a JSP page from automatically creating a session?

<%@ page session = "false">

How to pass information from JSP to included JSP?

Using <%jsp:param> tag.

Can we override the jspInit(), _jspService() and jspDestroy() methods?

We can override jspinit() and jspDestroy() methods but not _jspService().

Why is _jspService() method starting with an '_' while other life cycle methods do not?

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 an ''. This is the reason why we don't override _jspService() method in any JSP page.

A JSP page, *include.jsp* , has a instance variable "int a", now this page is statically included in another JSP page, home.jsp, which also has an instance variable "int a" declared. What happens when the home.jsp page is requested by the client?

It causes compilation error, as two variables with same name can't be declared. This happens because, when a page is included statically, entire code of included page becomes part of the new page. at this time there are two declarations of variable 'a'. Hence compilation error.

How is scripting disabled?

Scripting is disabled by setting the scripting-invalid element of the deployment descriptor to true. It is a subelement of jsp-property-group. Its valid values are true and false. The syntax for disabling scripting is as follows –

```
<jsp-property-group>
   <url-pattern>*.jsp</url-pattern>
   <scripting-invalid>true</scripting-invalid>
</jsp-property-group>
```

When do use application scope?

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

What are the options in JSP to include files?

In JSP, we can perform inclusion in the following ways –

By include directive – For example –

```
<%@ include file = ”header.jsp” %>
```

By include action – For example –

```
<%@ include file = ”header.jsp” %>
```

By using pageContext implicit object For example –

```
<% pageContext.include(“/header.jsp”); %>
```

By using RequestDispatcher object – For example –

```
<%
 RequestDispatcher rd = request.getRequestDispatcher(“/header.jsp”);
 Rd.include(request,response);
%>
```

How does JSP engines instantiate tag handler classes instances?

JSP engines will always instantiate a new tag handler instance every time a tag is encountered in a JSP page. A pool of tag instances are maintained and reusing them where possible. When a tag is encountered, the JSP engine will try to find a Tag instance that is not being used and use the same and then release it.

What’s the difference between JavaBeans and taglib directives?

JavaBeans and taglib fundamentals were introduced for reusability. But following are the major differences between them –

Taglibs are for generating presentation elements while JavaBeans are good for storing information and state.

Use custom tags to implement actions and JavaBeans to present information.

What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

Previous Page Next Page

Enter email for newsletter go