

# By OnlineInterviewQuestions.com

## JSP Interview Questions

**Find latest JSP Interview Questions and their answers**

**Q1. Which attribute specifies a JSP page that should process any exceptions thrown but not caught in the current page?**

**Q2. Explain What is JSP?**

**JSP**, expanded as **Java Server Pages** is a Java-based technology to create dynamic web pages. Released in 1999 by Sun Microsystems, JSP is based on technologies such as HTML, SOAP, XML, etc. JSP can be viewed as a servlet architecturally as it is translated into a servlet at runtime. JSP can also be used as a view component in MVC architecture with JavaBeans as a model and Servlets as the controller. JSP allows static content like HTML along with java code to be executed on the server-side. The resulting bytecode from the JSP must be executed on the JVM. In an overview, JSP can be thought of as an ASP or PHP technology to create web application but instead, JSP uses Java.

**Q3. What is the el in JSP?**

The **EL (Expression Language)** in the JSP is used to access the data stored in the JavaBeans along with implicit objects like requests, sessions, etc. It uses arithmetic and logical expressions to compute the accessed data. JSP supports data types such as integers, float, string, Boolean, and null.

### **Syntax**

```
${EL expression}
```

### **Example**

```
<jsp:setProperty name = "area" property = "parameter"
  areaOfCircle = "${3.14*area.radius*area.radius}"/> //here . is used to access
```

Expression Language supports most of the arithmetic and Logical operators for basic computation along with special operators like ‘.’ to access a property, ‘[]’ to access an array or list, and ‘()’ to group expression.

Functions can also be created using the JSP EL

## Syntax

```
#{ns:func(param1, param2, ...)} //ns is the function namespace and func is the
```

### Q4. How to print variables in JSP?

JSP uses the Java programming language. So, most of the statements in JSP follows the Java code. To print a variable in JSP,

```
<%  
String name=request.getParameter("Pname"); //initializing the variable name with  
out.print("welcome "+name); //statement to print the variable in JSP  
%>
```

### Q5. Explain OUT Implicit Object in JSP?

JSP has several implicit objects or predefined variables that developers can call without being explicitly declared. These are just Java objects and one such object is called Out implicit object.

The **out implicit object** is an instance of a JspWriter object that is used to send data or some content in response. This object can be declared based on the buffering of the page. The methods provided by this object are,

- **out.print(data type dt)** - it prints a data type value.
- **out.println(data type dt)** - it prints a data type value and terminates with a new line.
- **out.flush()** - it flushes the data stream.

### Q6. How to write comments in JSP?

The syntax for JSP comment is,

```
<%-- comment --%>
```

It tells the JSP container to ignore this part from the compilation as a comment.

```
<%--An Example for JSP comment--%>
```

### Q7. Enlist types of JSP tags?

Tags are an important part of JSP. There are 4 main types of JSP tags,

**Directive tag:** This type of tag is used to import packages into the current JSP application. The other use for this directive is to define error handling pages and for session information.

### Syntax

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

The three types of directive tags are,

- `<%@page...%>` - it defines page-dependent attributes.
- `<%@include...%>` - it includes a file.
- `<%@taglib...%>` - it declares a tag library to be used on the page.

### Example

```
<%@page language="java" session="true" %>  
<%@ include file="/title.jsp"%>
```

**Declaration tag:** This tag is used to declare variables or methods to be used in the Java code of the JSP.

### Syntax

```
<%! declaration; %>
```

### Example

```
<%! int k = 0; %>
```

**Scriptlet tag:** This tag contains the actual java code like java declaration, methods, expressions, etc.

### Syntax

```
<% java code %>
```

### Example

```
<% out.print("an example of scriptlet tag"); %>
```

**Expression tag:** The expression tag contains an expression that needs to be evaluated. It can contain any expression that is valid in Java code.

### Syntax

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

### Example

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

## Q8. What are directives in JSP?

The JSP directive provides information to the container on how to process the JSP page into its respective servlet.

### Syntax

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

There are three JSP directive attributes. They are,

- **<%@ page ... %>** - this is a page directive and it provides information about the JSP page like a scripting language, buffering requirements to the container. It has a number of attributes like buffer, autoFlush, contentType, session, etc to provide information.
- **<%@ include ... %>** - this is an include directive and it includes files pertaining to the translation phase.
- **<%@ taglib ... %>** - this is a tag library directive to define custom user-defined tags.

## Q9. How we use an expression in JSP?

**Expressions in the JSP** can be used in the expression tag to be evaluated. The expression tag converts the expression into a value of a string object and inserts it into the implicit output object to be displayed in the client browser. The semicolon is also no need to signal the end of the statement in the expression tag.

### Syntax

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

### Example

```
<%= "Expression Statement" %> //no need for output statement as the expression t
```

## Q10. What is the use of < Jsp: param> tag?

**JSP param tag** is one of the JSP action tags that is used to perform some specific tasks. The JSP param action is used to give information in the form of key/value pair. It can be used with JSP include, forward, and plugin action tag.

### Syntax

```
<jsp:param name="paramName" value="paramValue" />
```

### Example

```
<jsp:forward page="login.jsp" />  
//JSP param is used with the JSP forward action tag.  
</jsp:forward>
```

### Q11. Enlist different scope values for the <jsp:useBean> tag?

The **JSP useBean action tag** is used to locate the bean class is the bean object is instantiated or it instantiates the bean object. It has many attributes like id, scope, class, type, and beanName. The scope attribute in the useBean action tag is used to represent the scope of the bean. The default scope is the page, but different scopes are available such as request, session, and application.

- **Page** – The default scope that specifies the scope of the bean within the JSP page.
- **Request** – specifies the scope of the bean to all the JSP page that processes the same request.
- **Session** – specifies the scope of the bean to all the JSP page that has the same session irrespective of its request.
- **Application** – specifies the scope of the bean to all the JSP pages in the same application.

### Q12. Explain what is JSTL?

**JSTL (JSP Standard Tag Library)** is a collection of tags that brings valuable functionality to build JSP applications. According to their functions, JSTL is divided into 5 types.

- **Core tags** – this tag provides support for variables, URL management, and control flow by importing core libraries.
- **Function tags** – It provides support for string manipulation functions.
- **Formatting tags** – it provides formatting functions such as message formatting, date & time formatting, etc.
- **XML tags** – it is used to create and manipulate XML documents.
- **SQL tags** – As the name suggests, it provides SQL support to interact with databases.

### Q13. How to create and remove a session in JSP?

JSP Session uses HttpSession to create a session for each user. To create a session in JSP, we need to use setAttribute from the session implicit object.

#### Creating Session

```
session.setAttribute("user", name); //it creates the session for the object name
```

#### To remove the object from the session.

```
session.removeAttribute("user"); //deletes the session for the object name.
```

#### To remove the entire session.

```
session.invalidate(); - it kills the entire session and all the objects that are associated with this session.
```

#### Q14. What is the difference between JSP and JSPX?

**JSPX file** is nothing but an XML format for creating **JSP pages**. JSPX forces the separation of code from the view layer in different files. Javascript and XML are written in different files in JSPX whereas Java code and XML code are written in the same file in the JSP. Writing an XML format code is better in some cases than writing the native JSP. The XML code is easy to manipulate, errors can be rectified faster. Also, the JSPX file has a simpler syntax in XML than JSP pages. But, executing the logical or arithmetic expression in JSPX is difficult and hard. Dynamic content is easily generated on the JSP page than in the JSPX file.

#### Q15. What are Custom Tags in JSP? How do you create it?

The **custom tags** are just user-defined elements that are converted into some operation on an object when the JSP is translated into a servlet. Custom tags are instantiated from the SimpleTagSupport class. With this, scriptlet tag use is removed. And also, the custom tags can be reused.

**To create a custom tag in jsp, do the following steps.**

- Create the tag handler class and override the doStartTag() method.
- Create the TLD file which contains information about the tag and the tag handler class.
- Create the JSP file and use the created tag in it.

#### Q16. What is the use of the Include directive in JSP?

The **include directive** is used to include file to the JSP page during translation. These files can be an HTML, JSP, or normal text files. These files are useful in creating templates for user views. It is also used to break the pages into header, footer and sidebar sections.

Syntax

```
<%@ include...%>
```

Example

```
<%@ include file="header.jsp" %> //it includes header.jsp to the current file.
```

Please Visit [OnlineInterviewquestions.com](http://OnlineInterviewquestions.com) to download more pdfs