

# **JSP Assignment**

## **by**

Name: Prashant Maurya

EmpID : 12417

KIIT Roll : 1828259

What are the life-cycle methods for a JSP?

ANS-

### 1. Page Translation

A java servlet file is generated from the JSP source file. During the translation phase, the JSP engine reads the jsp file and parses it. The container validates the syntactic correctness of the JSP page and tag. By tags we mean standard directives and actions along with the custom actions referencing tag libraries used in the page.

### 2. Page Compilation

A java servlet file is generated from the JSP source file. During the translation phase, In this phase we compile the .java class of servlet into .class

### 3. Loading class and Instantiation

Once the class is compiled, the container loads the class and creates an instance of it.

### 4. JSP Initialization

Container calls `jspInit()` method to initialize the servlet instance before serving any request. This method is called only once for a servlet and is usually used for one time activity like initializing database connections etc.

SYNTAX:-

```
public void jspInit(){  
    // Initialization code...  
}
```

### 5. JSP Execution

This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed. 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 on request basis. This 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, i.e, GET, POST, DELETE, etc.

SYNTAX:-

```
void _jspService(HttpServletRequest request, HttpServletResponse response) {  
    // Service handling code...  
}
```

### 6. JSP 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. Override `jspDestroy` when you need to perform any cleanup, such as releasing database connections or closing open files.

SYNTAX:-

```
public void jspDestroy() {  
    // Your cleanup code goes here.  
}
```

=====

## 2) What is JSP?

ANS- JavaServer Pages (JSP) is a technology for developing Web pages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags.

=====

## 3) What are the Literals used in JSP?

ANS-

The following literals are used in JSP-

1. Boolean – true and false
2. Integer – as in Java
3. Floating point – as in Java
4. String – with single and double quotes; " is escaped as \", ' is escaped as \', and \ is escaped as \\\.
5. Null – null

=====

## 4) List down the major differences between the JSP Custom Tags and Java Beans.

ANS-

Custom JSP tags-

1. Can manipulate JSP content
2. Custom tags can simplify the complex operations much better than the bean can. But require a bit more work to set up.
3. Used only in JSPs in a relatively self-contained manner. Can be used in both Servlets and JSPs.

Java Beans-

1. Can't manipulate JSP content.
2. Easier to set up

3.You can define a bean in one Servlet and use them in another Servlet or a JSP page.

=====

5) What are the ways to include the result of another page. ?

ANS-There are two ways to include the result of another page:

1.By include directive

2.By include action

=====

6) What is the difference between include directive and include action?

ANS-

Include Directive

1) The include directive includes the content at page translation time. 2) The include directive includes the original content of the page, so page size increases at runtime

3) It's better for static pages.

Include Action

1) The include action includes the content at request time.

2) The include action doesn't include the original content rather invokes the include() method of Vendor provided class.

3) It's better for dynamic pages.

=====

7) Can an interface be implemented in the JSP file? True or False ?

Solution:-

False , an interface can't be implemented in the jsp file but we can use a javabean in a jsp and this javabean can implement interface

=====

8) Create an JSP page where you perform an operation of adding two numbers and displaying it between the HTML's

tag

Sol

html->

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
</head>
```

```
<body>

    <form action="added.jsp">

        <div>enter 1st number: <input type="text" name="num1"></div>
        <hr>
        <div>enter 2nd number: <input type="text" name="num2"></div>
        <hr>
        <input type="submit">

    </form>

</body>
</html>
```

Jsp page->

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Adding</title>
</head>
<body>
    <h1>Add Two Number</h1>
    <p>
        <%

            int i1 = Integer.parseInt(request.getParameter("num1"));
            int i2 = Integer.parseInt(request.getParameter("num2"));

            int k = i1+i2;

            out.println("k is "+k);

        %>
    </p>

</body>
</html>
```

=====

9) Create an JSP application where you create a form page where you submit some form data(you decide what field you want to use) after submitting the form. Create two jsp pages one which will be executed once the form is submitted and then forward those formdata to another jsp page and display it over there.

Sol

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

    <form action="added.jsp">

        <div>enter 1st number: <input type="text" name="num1"></div>
        <hr>
        <div>enter 2nd number: <input type="text" name="num2"></div>
        <hr>
        <input type="submit">

    </form>

</body>
</html>
```

```
1 jsp->
<%@page import="org.apache.jasper.compiler.Node.ForwardAction"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Adding</title>
</head>
<body>
    <h1>Add Two Number</h1>
    <p>
    <%

        int i1 = Integer.parseInt(request.getParameter("num1"));
```

```

    int i2 = Integer.parseInt(request.getParameter("num2"));

    int k = i1+i2;

    request.setAttribute("sum", k);
    RequestDispatcher rd = request.getRequestDispatcher("squarelt.jsp");
    rd.forward(request, response);

    %>
</p>

</body>
</html>

```

2nd jsp->

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

```

```

    <h3>
    catching data to sq it->>

    <%

```

```

        int k = Integer.parseInt(request.getParameter("sum"));

        out.println(k+" is sum "+ k*k + " is sq");

    %>

    </h3>

```

```

</body>
</html>

```

