Question 1- Introduction to JSP

**Introduction**

* 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.

**JSP pages are more advantageous than Servlet:**

* They are easy to maintain.
* No recompilation or redeployment is required.
* JSP has access to entire API of JAVA .
* JSP are extended version of Servlet.

**Features of JSP**

* **Coding in JSP is easy** :- As it is just adding JAVA code to HTML/XML.
* **Reduction in the length of Code** :- In JSP we use action tags, custom tags etc.
* **Connection to Database is easier** :-It is easier to connect website to database and allows to read or write data easily to the database.
* **Make Interactive websites** :- In this we can create dynamic web pages which helps user to interact in real time environment.
* **Portable, Powerful, flexible and easy to maintain** :- as these are browser and server independent.
* **No Redeployment and No Re-Compilation** :- It is dynamic, secure and platform independent so no need to re-compilation.
* **Extension to Servlet** :- as it has all features of servlets, implicit objects and custom tags
* **JSP syntax**
* Syntax available in JSP are following
* **Declaration Tag** :-It is used to declare variables.

**Syntax:-**

<%! Dec var %>

**Example:-**

<%! int var=10; %>

* **Java Scriplets** :- It allows us to add any number of JAVA code, variables and expressions.

**Syntax:-**

<% java code %>

* **JSP Expression** :- It evaluates and convert the expression to a string.

**Syntax:-**

<%= expression %>

**Example:-**

<% num1 = num1+num2 %>

* **JAVA Comments** :- It contains the text that is added for information which has to be ignored.

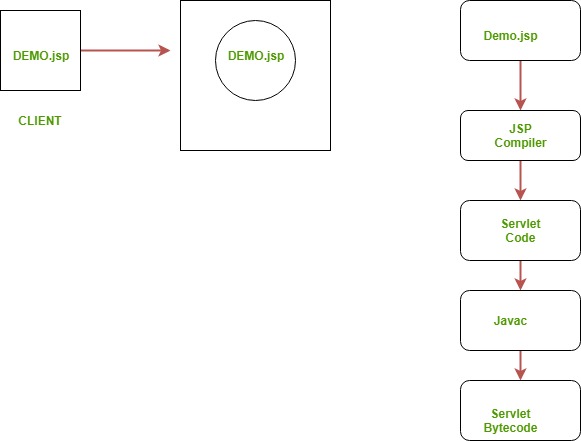
**Syntax:-**

<% -- JSP Comments %>

**Process of Execution**

Steps for Execution of JSP are following:-

* + Create html page from where request will be sent to server eg try.html.
  + To handle to request of user next is to create .jsp file Eg. new.jsp
  + Create project folder structure.
  + Create XML file eg my.xml.
  + Create WAR file.
  + Start Tomcat
  + Run Application



**Example of Hello World**

We will make one .html file and .jsp file

**demo.jsp**

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Hello World - JSP tutorial</title>

</head>

<body>

<%= "Hello World!" %>

</body>

</html>

**Advantages of using JSP**

* + It does not require advanced knowledge of JAVA
  + It is capable of handling exceptions
  + Easy to use and learn
  + It contains tags which are easy to use and understand
  + Implicit objects are there which reduces the length of code
  + It is suitable for both JAVA and non JAVA programmer

**Disadvantages of using JSP**

* + Difficult to debug for errors.
  + First time access leads to wastage of time
  + It’s output is HTML which lacks features.

### Question 2- Creating a simple JSP Page

To create the first JSP page, write some HTML code as given below, and save it by .jsp extension. We have saved this file as index.jsp. Put it in a folder and paste the folder in the web-apps directory in apache tomcat to run the JSP page.

**index.jsp**

Let's see the simple example of JSP where we are using the scriptlet tag to put Java code in the JSP page. We will learn scriptlet tag later.

1. <html>
2. <body>
3. <% out.print(2\*5); %>
4. </body>
5. </html>

Question 2-Introduction of servlet

Servlets provide a component-based, platform-independent method for building Webbased applications, without the performance limitations of CGI programs. Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. This tutorial will teach you how to use Java Servlets to develop your web based applications in simple and easy steps.

## Why to Learn Servlet?

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI.

* Performance is significantly better.
* Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
* Servlets are platform-independent because they are written in Java.
* Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted.
* The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

## Applications of Servlet

* Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
* Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
* Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
* Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.
* Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks.

## Prerequisites

We assume you have good understanding of the Java programming language. It will be great if you have a basic understanding of web application and how internet works.

# Question 3- JSTL (JSP Standard Tag Library)

The JSP Standard Tag Library (JSTL) represents a set of tags to simplify the JSP development.

## Advantage of JSTL

1. **Fast Development** JSTL provides many tags that simplify the JSP.
2. **Code Reusability** We can use the JSTL tags on various pages.
3. **No need to use scriptlet tag** It avoids the use of scriptlet tag.

## JSTL Tags

There JSTL mainly provides five types of tags:

|  |  |
| --- | --- |
| **Tag Name** | **Description** |
| [Core tags](https://www.javatpoint.com/jstl-core-tags) | The JSTL core tag provide variable support, URL management, flow control, etc. The URL for the core tag is **http://java.sun.com/jsp/jstl/core**. The prefix of core tag is **c**. |
| [Function tags](https://www.javatpoint.com/jstl-function-tags) | The functions tags provide support for string manipulation and string length. The URL for the functions tags is **http://java.sun.com/jsp/jstl/functions** and prefix is **fn**. |
| [Formatting tags](https://www.javatpoint.com/jstl-formatting-tags) | The Formatting tags provide support for message formatting, number and date formatting, etc. The URL for the Formatting tags is **http://java.sun.com/jsp/jstl/fmt** and prefix is **fmt**. |
| [XML tags](https://www.javatpoint.com/jstl-xml-tags) | The XML tags provide flow control, transformation, etc. The URL for the XML tags is **http://java.sun.com/jsp/jstl/xml** and prefix is **x**. |
| [SQL tags](https://www.javatpoint.com/jstl-sql-tags) | The JSTL SQL tags provide SQL support. The URL for the SQL tags is **http://java.sun.com/jsp/jstl/sql** and prefix is **sql**. |

Question 4- Life Cycle of a Servlet (Servlet Life Cycle)

The web container maintains the life cycle of a servlet instance. Let's see the life cycle of the servlet:

1. Servlet class is loaded.
2. Servlet instance is created.
3. init method is invoked.
4. service method is invoked.
5. destroy method is invoked.



As displayed in the above diagram, there are three states of a servlet: new, ready and end. The servlet is in new state if servlet instance is created. After invoking the init() method, Servlet comes in the ready state. In the ready state, servlet performs all the tasks. When the web container invokes the destroy() method, it shifts to the end state.