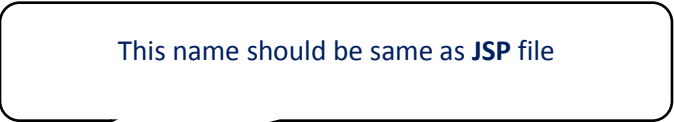

Steps to create a simple web application using JSP

Step 1: create a HTML page containing a **<form>** element.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Factorial</title>
</head>
<body>
  <form action="Calculate.jsp" method="post">
    <table>
      <tr>
        <td>Enter a number:</td>
        <td><input type="text" name="fact"></td>
      </tr>
      <tr>
        <td colspan="2"><input type="submit" value="find"/></td>
      </tr>
    </table>
  </form>
</body>
</html>
```



Step 2: create a new **jsp** file ex: **Calculate.jsp**.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Factorial Calculator</title>
</head>
<body>

</body>
</html>
```

Step 3: Use the **declarative tag** **<%! %>** to declare and initialize the required instance variables and methods.

- a. Whatever placed inside these tags gets initialized during JSP initialization phase and has class scope.

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Factorial Calulator</title>
</head>
<body>
    <%! int fact = 1; %> <!-- Declarative Tag -->
</body>
</html>

```

Step 4: Use the **scriptlet tag** `<% %>` to perform the data processing. Use the implicit **request** object to read the form data.

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Factorial Calulator</title>
</head>
<body>
    <!-- Declarative Tag -->
    <%! int fact = 1; %>

    <!-- Scriptlet Tag -->
    <%
        long number = Long.parseLong(request.getParameter("fact"));
        for (int i = 1; i <= number; i++) {
            fact = fact * i;
        }
    %>

</body>
</html>

```

Step 5: Use the **expression tag** `<%= %>` to print the data.

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Factorial Calculator</title>
</head>
<body>
    <!-- Declarative Tag -->

    <%! int fact = 1;%>

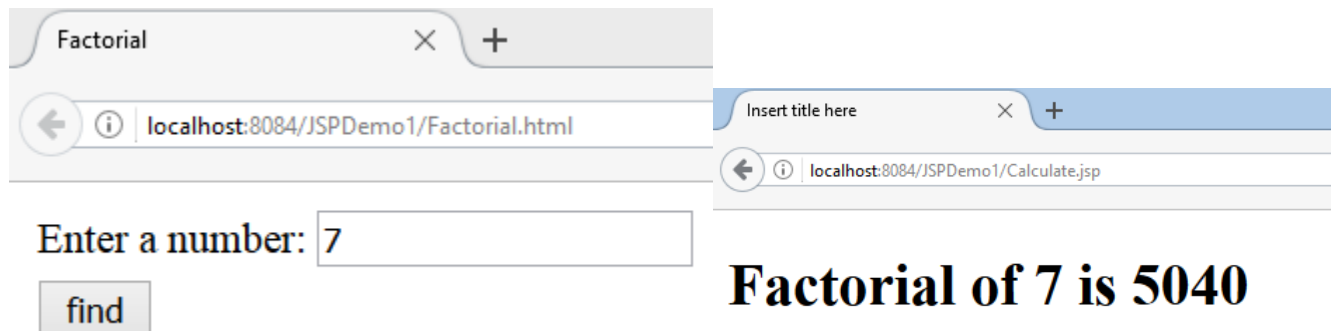
    <!-- Scriptlet Tag -->
    <%
        long number = Long.parseLong(request.getParameter("fact"));
        for (int i = 1; i <= number; i++) {
            fact = fact * i;
        }
    %>

    <!-- Expression Tag -->
    <h1>Factorial of <%=number%> is <%=fact%></h1>

</body>
</html>

```

Step 6: *Run the HTML file.*



Steps to create a web application using JSP for Session Management

Step 1: create a HTML page containing a **<form>** element.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Login</title>
</head>
<body>
  <FORM NAME="login" ACTION="LoginSession.jsp" METHOD="POST">
    <table>
      <tr>
        <td>Enter username :</td>
        <td><input type="text" name="username"/></td>
      </tr>
      <tr>
        <td>Enter Password :</td>
        <td><input type="password" name="pass"/></td>
      </tr>
      <tr>
        <td><input type="submit" value="Submit" /></td>
        <td><input type="reset" value="Clear" /></td>
      </tr>
    </table>
  </FORM>
</body>
</html>
```

Step 2: create a new **JSP** file and read the parameters using the **request** object.

```
<%@page import="sun.security.util.Password"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login Session</title>
</head>
<body>
```

```

    <%
        String username = request.getParameter("username");
        String pass = request.getParameter("pass");

    %>
</body>
</html>
</html>

```

Step 3: Use the *session* object to set the session attribute and redirect to another page
ex: *Success.jsp*.

```

<%@page import="sun.security.util.Password"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login Session</title>
</head>
<body>
    <%
        String username = request.getParameter("username");
        String pass = request.getParameter("pass");

        if (pass.equals("admin")) {
            session.setAttribute("user", username);
            response.sendRedirect("Success.jsp");
        } else {
            session.setAttribute("msg", "Invalid login credentials");
            response.sendRedirect("Error.jsp");
        }

    %>
</body>
</html>
</html>

```

Step 4: Retrieve the session attribute in the *Success.jsp* page and print it using the expression tag `<%= %>`.

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login Success</title>
</head>
<body>
<h1>Login Success!</h1>
<h2>Welcome <%=session.getAttribute("user") %></h2>
</body>
</html>

```

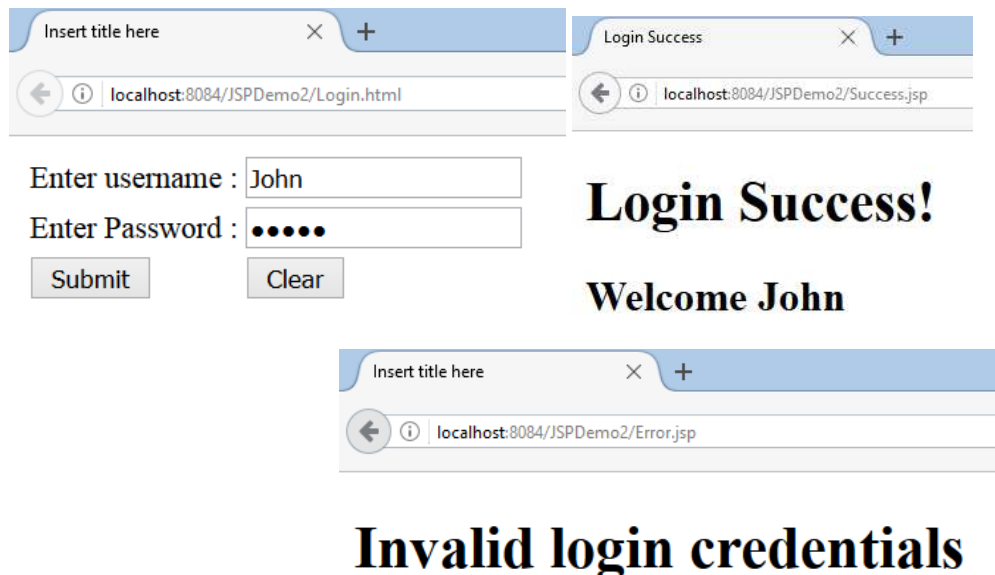
Step 5: *Display any error messages in Error.jsp*

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1><%=session.getAttribute("msg") %></h1>
</body>
</html>

```

Step 6: *Run the HTML file*



Steps to create a web application using JSP for Java Beans

Step 1: Create a **java bean/entity** class in the **src** folder. A java bean class is a class that contains only the instance variables, constructors and the getter and setter methods

```
package com.wipro.entity;

public class Car {

    private int carId;
    private String name;
    private double price;

    public int getCarId() {
        return carId;
    }

    public void setCarId(int carId) {
        this.carId = carId;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

}
```

Step 2: create a HTML page containing a **<form>** element.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Cars</title>
</head>
<body>
  <form name="carForm" action="CarRegister.jsp" method="post">
    <table>
      <tr>
        <td>Car Id:</td>
        <td><input type="text" name="carId" /></td>
      </tr>
      <tr>
        <td>Car Name:</td>
        <td><input type="text" name="name" /></td>
      </tr>
      <tr>
        <td>Car Price:</td>
        <td><input type="text" name="price" /></td>
      </tr>
      <tr>
        <td colspan="2"><input type="submit" value="Register" /></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

The name of each HTML element must be same as given in Bean class

Step 3: create a JSP file to process the form data.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Car Register</title>
</head>
<body>

</body>
</html>
```


Step 4: Use the `<jsp:useBean/>` tag to create an instance of bean class.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Car Register</title>
</head>
<body>
    <jsp:useBean id="carObj" class="com.wipro.entity.Car" />
</body>
</html>
```



The object of bean class

The bean class name

Step 5: Use the `<jsp:setProperty/>` tag to initialize the properties of bean class with the data given in HTML form.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Car Register</title>
</head>
<body>
    <jsp:useBean id="carObj" class="com.wipro.entity.Car" />

    <jsp:setProperty name="carObj" property="carId" />
    <jsp:setProperty name="carObj" property="name" />
    <jsp:setProperty name="carObj" property="price" />

</body>
</html>
```



The object of bean class

The bean class properties. Should be same as given in HTML form

Step 6: Use the `<jsp:getProperty/>` tag to retrieve the assigned values.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Car Register</title>
</head>
<body>
    <jsp:useBean id="carObj" class="com.wipro.entity.Car" />
    <jsp:setProperty name="carObj" property="carId" />
    <jsp:setProperty name="carObj" property="name" />
    <jsp:setProperty name="carObj" property="price" />

    <h3>The car details are</h3>
    car id:<jsp:getProperty name="carObj" property="carId" /><br>
    car name:<jsp:getProperty name="carObj" property="name" /><br>
    car price:Rs.<jsp:getProperty name="carObj" property="price" />
</body>
</html>
```

Step 7: Run the HTML file

The image shows two side-by-side browser windows. The left window, titled 'Cars', displays a form with three input fields: 'Car Id:' with value '1001', 'Car Name:' with value 'Honda City', and 'Car Price:' with value '800000'. Below these fields is a 'Register' button. The right window, titled 'Car Register', shows the output of the JSP page. It displays the text 'The car details are' followed by three lines: 'car id:1001', 'car name:Honda City', and 'car price:Rs.800000.0'.

Car Id:	1001
Car Name:	Honda City
Car Price:	800000

Register

The car details are

car id:1001
car name:Honda City
car price:Rs.800000.0

JSP Action Tags

- I. ***jsp:forward*** - forwards the request and response to another resource.

Syntax of jsp:forward action tag without parameter

```
<jsp:forward page="relativeURL | <%= expression %>" />
```

Syntax of jsp:forward action tag with parameter

```
<jsp:forward page="relativeURL | <%= expression %>">  
<jsp:param name="parametername" value="parametervalue | <%=expression%>" />  
</jsp:forward>
```

Examples:

- 1) `<jsp:forward page="Home.jsp" />`
- 2) `<jsp:forward page="Home.jsp" >`
 `<jsp:param name="name" value="John" />`
 `</jsp:forward>`

- II. ***jsp:include*** - includes another resource.

Syntax of jsp:include action tag without parameter

```
<jsp:include page="relativeURL | <%= expression %>" />
```

Syntax of jsp:include action tag with parameter

```
<jsp:include page="relativeURL | <%= expression %>">  
<jsp:param name="parametername" value="parametervalue | <%=expression%>" />  
</jsp:include>
```

Examples:

- 1) `<jsp:include page="Home.jsp" />`
- 2) `<jsp:include page="Home.jsp" >`
 `<jsp:param name="name" value="John" />`
 `</jsp:include>`

III. *jsp:useBean* - creates or locates bean object.

Syntax of *jsp:useBean* action tag

```
<jsp:useBean id="instanceName" scope="page | request | session | application"  
class="packageName.className" type="packageName.className"  
beanName="packageName.className | <%= expression >" >  
</jsp:useBean>
```

- **id:** is used to identify the bean in the specified scope.
- **scope:** represents the scope of the bean. It may be page, request, session or application.

The default scope is page.

- **page:** specifies that you can use this bean within the JSP page. The default scope is page.
 - **request:** specifies that you can use this bean from any JSP page that processes the same request. It has wider scope than page.
 - **session:** specifies that you can use this bean from any JSP page in the same session whether processes the same request or not. It has wider scope than request.
 - **application:** specifies that you can use this bean from any JSP page in the same application. It has wider scope than session.
- **class:** instantiates the specified bean class (i.e. creates an object of the bean class) but it must have no-arg or no constructor and must not be abstract.
 - **type:** provides the bean a data type if the bean already exists in the scope. It is mainly used with class or beanName attribute. If you use it without class or beanName, no bean is instantiated.
 - **beanName:** instantiates the bean using the `java.beans.Beans.instantiate()` method.

Examples:

1) `<jsp:useBean id="obj" class="com.mypackage.MyClass"/>`

IV. ***jsp:setProperty*** - Sets the property of a JavaBean.

Syntax of jsp:setProperty action tag

```
<jsp:setProperty name="instanceOfBean" property="*" |  
property="propertyName" param="parameterName" |  
property="propertyName" value="{ string | <%= expression %>}"  
/>
```

Examples:

To set all the values of incoming request in the bean

```
<jsp:setProperty name="beanObj" property="*" />
```

To set value of the incoming specific property

```
<jsp:setProperty name="beanObj" property="username" />
```

To set a specific value in the property

- `<jsp:setProperty name="beanObj" property="username" value="John" />`
- `<jsp:setProperty name="beanObj" property="username" value="<%=
request.getParameter("EmployeeId") %>" />`
- `<jsp:setProperty name="beanObj" property="username" param="uname" />`

V. ***jsp:getProperty*** - returns the value of the property.

Syntax of jsp:getProperty action tag

```
<jsp:getProperty name="instanceOfBean" property="propertyName" />
```

Examples:

```
<jsp:getProperty name=" beanObj " property="username" />
```