

# JSP - Java Server Page

## 1. Một số chú ý

- **Đầu tệp (pageEncoding="utf-8" – nếu tiếng Việt)**

```
<% @page contentType="text/html" pageEncoding="utf-8"%>
```

- **Mã JSP (Scriptlets):**

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

Ví dụ viết vào văn bản:

```
<% out.println("Hello JSP!"); %>
```

```
<% out.write("<h2>Let start it!</h2>");%>
```

- **Khai báo (Declarations):**

```
<%! int x=1; %>
```

- **Biểu thức (Expressions):**

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

- **Chú thích (comment): đâu là chú thích??**

*Syntax: chú thích mã HTML*

```
<!-- comment [ <%= expression %> ] -->
```

– Example:

```
<!-- This page was loaded on
```

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

```
-->
```

### Bảng sau liệt kê 9 đối tượng ẩn mà JSP hỗ trợ:

No.	Đối tượng & Mô tả
1	<b>request</b> Đây là đối tượng <b>HttpServletRequest</b> được liên kết với request.
2	<b>response</b> Đây là đối tượng <b>HttpServletResponse</b> được liên kết với response tới client.
3	<b>out</b> Đây là đối tượng <b>PrintWriter</b> được sử dụng để gửi dữ liệu tới client.
4	<b>session</b> Đây là đối tượng <b>HttpSession</b> được liên kết với request.
5	<b>application</b> Đây là đối tượng <b>ServletContext</b> được liên kết với application context.
6	<b>config</b> Đây là đối tượng <b>ServletConfig</b> được liên kết với page.
7	<b>pageContext</b> Điều này đóng gói việc sử dụng các tính năng cụ thể của server như <b>JspWriters</b> với hiệu suất cao hơn.
8	<b>page</b> Đơn giản là một từ đồng nghĩa với <b>this</b> , được sử dụng để gọi các phương thức được định nghĩa bởi lớp ser được biên dịch.
9	<b>exception</b> Đối tượng <b>exception</b> cho phép các dữ liệu ngoại lệ (Exception) được truy cập bằng JSP được chỉ định.

## 2. Các thẻ điều hướng

Có 3 loại thẻ **directive** trong JSP:

Directive	Miêu tả
-----------	---------

<code>&lt;% @ page ... %&gt;</code>	Định nghĩa một thuộc tính page-dependent (phụ thuộc trang), như ngôn ngữ scripting, trang lỗi và các yêu cầu bộ đệm
-------------------------------------	---

<code>&lt;% @ include ... %&gt;</code>	Include một file trong suốt giai đoạn biên dịch
<code>&lt;% @ taglib ... %&gt;</code>	Khai báo một thư viện thẻ, chứa các action tùy biến, được sử dụng trong trang đó

**2.1 page directive** được sử dụng để cung cấp các chỉ lệnh tới container mà liên quan tới JSP page hiện tại. Ta có thể mã hóa page directive bất cứ đâu trong JSP page.

Cú pháp:

`<% @page attribute="value" %>` Ví dụ:

`<% @ page buffer="none" %>`

`<% @ page buffer="8kb" %>`

`<% @page contentType="text/html" pageEncoding="UTF-8"%>`

`<% @ page import="java.sql.*" %>`

`<% @page import="java.util.Date, java.text.SimpleDateFormat"%>`

**2.2 include Directive** trong JSP được sử dụng để bao một file trong suốt giai đoạn biên dịch. Ta có thể mã hóa **include** directive bất cứ đâu trong JSP page.

`<% @ include`

`file="relative url" >`

Ví dụ:

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

`<% @include file="logo.html" %>`

`<% @include file="footer.jsp" %>`

Java Server Pages API cho phép định nghĩa các Custom Tags trong JSP, mà giống như các thẻ HTML hoặc XML và một thư viện thẻ (tag library) là một tập hợp các thẻ được định nghĩa bởi người sử dụng mà triển khai các

hành vi tùy biến.

**2.3 taglib directive** khai báo rằng: JSP page sử dụng một tập hợp các Custom Tags, thông báo vị trí của thư viện, và cung cấp một phương thức để nhận diện các Custom Tags đó trong JSP page.

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

## 2.4 JSP Action Tags

JSP action sử dụng các chỉ lệnh trong cú pháp XML để điều khiển hành vi của Servlet engine. Có thể chèn động một file, tái sử dụng các thành phần JavaBeans, chuyển người dùng tới trang khác, hoặc tạo HTML cho Java plugin.

JSP Action Tags	Description
jsp:forward	forwards the request and response to another resource.
jsp:include	includes another resource.
jsp:useBean	creates or locates bean object.
jsp:setProperty	sets the value of property in bean object.
jsp:getProperty	prints the value of property of the bean.
jsp:plugin	embeds another components such as applet.
jsp:param	sets the parameter value. It is used in forward and include mostly.
jsp:fallback	can be used to print the message if plugin is working. It is used in jsp:plugin.

**forward** action trong JSP kết thúc action của trang hiện tại và chuyển request tới nguồn khác chẳng hạn như một Static page, JSP page khác, hoặc một Java Servlet.

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

<jsp:include> Action trong JSP

Action này cho phép bạn chèn các file vào trong các trang đang được tạo. Cú pháp như sau:

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

### 3. Java bean:

**<jsp:useBean>** Action trong JSP

useBean action trong JSP là khá linh hoạt. Đầu tiên nó tìm kiếm một đối tượng đang tồn tại đang sử dụng thuộc tính id và scope. Nếu một đối tượng không được tìm thấy, thì nó cố gắng tạo đối tượng đã xác định đó.

Một JavaBean là một **lớp Java** được xây dựng một cách đặc biệt trong Java và được mã hóa theo JavaBeans API Specifications.

Sau đây là các đặc trưng duy nhất giúp phân biệt một JavaBean với các lớp khác trong Java:

Nó cung cấp một constructor mặc định, không có tham số.

Nó có thể xếp thứ tự và triển khai **Serializable** interface.

Nó có thể có một số đặc tính mà có thể được đọc và được viết.

Nó có thể có một số phương thức "getter" và "setter" cho các đặc tính.

**Cách đơn giản nhất để tải một Bean như sau:**

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

Trong đó scope="page" là mặc định, ngoài ra “application”, “session” hoặc “request”

```
<jsp:useBean id="students" class = "bean.StudentsBean"/>
```

Tương đương

```
<% bean.StudentsBean students = new bean.StudentsBean(); %>
```

Khi lớp Bean đã được tải, có thể sử dụng các <jsp:setProperty> action và

<jsp:getProperty> action để sửa đổi và thu nhận các đặc tính của Bean.

```
<jsp:getProperty name="students" property="firstName"/>
```

Tương đương <%= students.getFirstName() %>

```
<jsp:getProperty name="students" property="*/>
```

```
<jsp:setProperty name="students" property="firstName" value="Loan"/>
```

Tương đương <%= students.setFirstName(“Loan”) %>

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

## Expression Language (EL) trong JSP

Expression Language (EL) cho phép JSP developers truy cập các đối tượng java thông qua tag. JSP EL cho phép chúng ta tạo các Expression, gồm số học và logic. Bên trong một JSP EL, bạn có thể sử dụng các integer, các số floating point, string, các hằng có sẵn true hoặc false cho các giá trị Boolean, và null.

### JavaBean

Một JavaBean là một lớp Java được xây dựng một cách đặc biệt trong Java và được mã hóa theo JavaBeans API Specifications.

Sau đây là các đặc trưng duy nhất giúp phân biệt một JavaBean với các lớp khác trong Java:

- Nó cung cấp một constructor mặc định, không có tham số.
- Nó có thể xếp thứ tự và triển khai Serializable interface.
- Nó có thể có một số đặc tính mà có thể được đọc và được viết.
- Nó có thể có một số phương thức "getter" và "setter" cho các đặc tính.

Đọc tại

[https://www.tutorialspoint.com/jsp/jsp\\_java\\_bean](https://www.tutorialspoint.com/jsp/jsp_java_bean)

s.htm Ví dụ:

```
public class Student{
    private int rollnum;
    private String name;
    public Student() {
    }
    public Student(int rollnum, String name) {
        this.rollnum = rollnum;
        this.name = name;
    }
    public int getRollnum() {
        return rollnum;
    }
    public void setRollnum(int rollnum) {
        this.rollnum = rollnum;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

```
}
```

## File JSP:

```
<form action="display.jsp" method="post">
    Enter ur roll number:
    <input type="number" name="rollnum"/><br/>
    Enter ur name:
    <input type="text" name="name"/><br/>
    <input type="submit" value="Submit">
</form>
```

## Khai báo sử dụng EL

```
<%@page isELIgnored="false" %>
```

## File display.jsp

<pre>//scriptlet -jsp &lt;%     Entity.Student s1=new         Student(); int rollnum = Integer.parseInt(request.getParameter("rollnu m")); String name= request.getParameter("name");     Student s=new Student(rollnum,         name);     %&gt;     Roll number:     &lt;%=s.getRollnum()%&gt;&lt;br/&gt;     Name:&lt;%=s.getName()%&gt;&lt;br/&gt;</pre>	<pre>&lt;jsp:useBean id="s1" class="entity.Student" scope="request"&gt; &lt;/jsp:useBean&gt;     &lt;jsp:setProperty name="s1" property="*/&gt;         Roll number:&lt;jsp:getProperty name="s1" property="rollnum"/&gt;&lt;br/&gt;         Name:&lt;jsp:getProperty name="s1" property="name"/&gt;&lt;br/&gt;      &lt;jsp:useBean id="s2" class="entity.Student" scope="request"/&gt;     &lt;jsp:setProperty name="s2" property="rollnum" value="\${param.rollnum}"/&gt;     &gt;     &lt;jsp:setProperty name="s2" property="name" value="\${param.name}"/&gt;         Roll number:&lt;jsp:getProperty name="s2" property="rollnum"/&gt;&lt;br/&gt;         Name:&lt;jsp:getProperty name="s2" property="name"/&gt;&lt;br/&gt;</pre>
--	---

## Expression Language

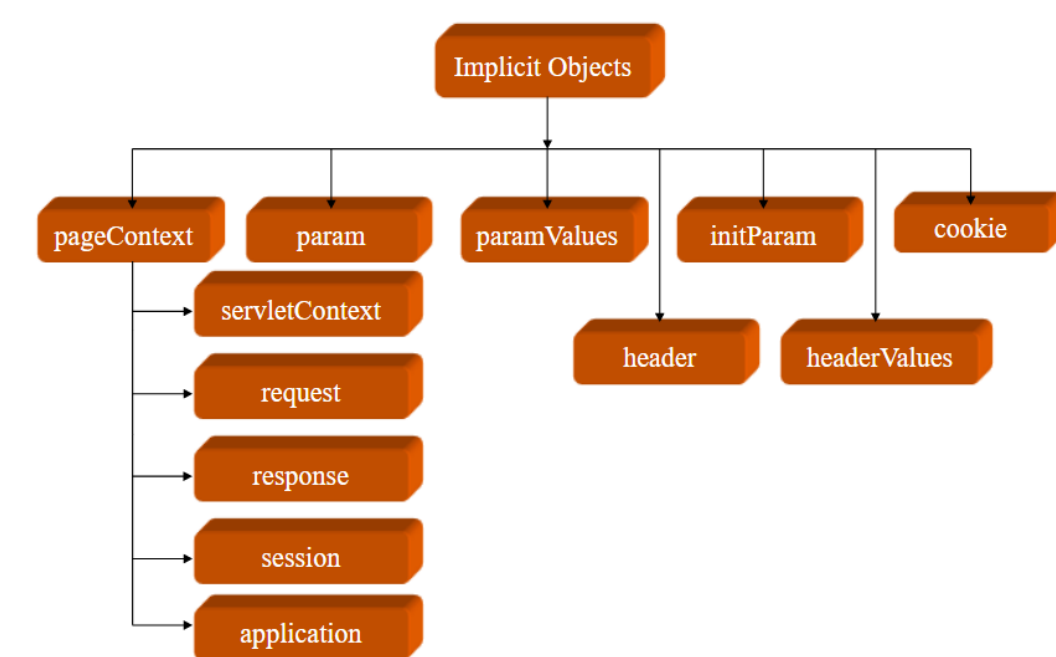
Roll number: \${param["rollnum"]}   - \${param.rollnum} Name: \${param.name} <%=request.getParameter("name")%>
---

### 1. Các đối tượng ẩn trong JSP EL

JSP Expression Language hỗ trợ các đối tượng ẩn sau:

Đối tượng ẩn	Miêu tả
pageScope	Các biến scope từ phạm vi page
requestScope	Các biến scope từ phạm vi request
sessionScope	Các biến scope từ phạm vi session
applicationScope	Các biến scope từ phạm vi application
param	Các tham số request, dạng các chuỗi
paramValues	Các tham số request, dạng các tập hợp của các chuỗi
header	HTTP request header dạng các chuỗi
headerValues	HTTP request header dạng các tập hợp của các chuỗi
initParam	Các tham số khởi tạo context
cookie	Các giá trị Cookie
pageContext	Đối tượng JSP PageContext cho page hiện tại





## 2. Toán tử cơ bản trong EL

JSP Expression Language (EL) hỗ trợ hầu hết các toán tử số học và logic được hỗ trợ bởi Java. Dưới đây là danh sách các toán tử được sử dụng thường xuyên nhất.

Toán tử	Miêu tả
.	Truy cập một đặc tính của Bean hoặc Map entry
[]	Truy cập một phân tử mảng hoặc List
( )	Nhóm một subexpression để thay đổi thứ tự ước lượng
+	Phép cộng
-	Phép trừ hoặc phủ định một giá trị
*	Phép nhân
/ hoặc div	Phép chia
% hoặc mod	Phép chia lấy phần dư
== hoặc eq	Kiểm tra có bằng hay không
!= hoặc ne	Kiểm tra tính không bằng

< hoặc lt	Kiểm tra tính nhỏ hơn
> hoặc gt	Kiểm tra tính lớn hơn
<= hoặc le	Kiểm tra tính nhỏ hơn hoặc bằng
>= hoặc ge	Kiểm tra tính lớn hơn hoặc bằng
&& hoặc and	Phép AND logic
hoặc or	Phép OR logic
! hoặc not	Phản bù Boolean một ngôi
empty	Kiểm tra các giá trị biến rỗng

### Example 1: Calculate

```

<h1>Calculating with EL</h1>
  <form method="post">
    Enter number 1:
    <input type="text" name="num1" value="{param.num1}"/><br/>
    Enter number 2:
    <input type="text" name="num2" value="{param.num2}"/><br/>

    <h3 style="color: blue">Sum: {param.num1 + param.num2}</h3>
    <h3 style="color: blue">Avg: {(param["num1"] + param["num2"])/2}</h3>
  <input type="submit" value="Calculate"/>
</form>

```

# JSTL (JSP Standard Tag Library)

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

## Classification of The JSTL Tags

The JSTL tags can be classified, according to their functions, into the following JSTL tag library groups that can be used when creating a JSP page –

- **Core Tags**
- **Formatting tags**
- **SQL tags**
- **XML tags**
- **JSTL Functions**

The JSTL core tag provides variable support, URL management, flow control etc. The syntax used for including JSTL core library in your JSP is:

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

### Example : <c:set>

```
<c:set var="n" value="10"/>
```

```
<c:set var="id" value="${requestScope.id}"/>
```

### Example 2: <c:forEach>

#### • Servlet

```
List<Student> data=new ArrayList<>();
data.add(new Student(1, "Nguyen Van A"));
data.add(new Student(2, "Tran Thi B"));
data.add(new Student(3, "Vu Van C"));
data.add(new Student(4, "Tran thi Ty"));
request.setAttribute("data", data);
request.getRequestDispatcher("list.jsp").forward(request, response);
```

#### • JSP

```
<%@page isELIgnored="false" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

<pre>&lt;c:forEach items="\${requestScope.data}" var="s"&gt;     \${s.rollnum} \${s.name}&lt;br/&gt; &lt;/c:forEach&gt;</pre>	<pre>\${requestScope["data"]}</pre>
---	-------------------------------------

```
<c:forEach begin="${1}" end="${10}" var="i">
    ${i}<br/>
</c:forEach>
```

Servlet

```
HttpSession session = request.getSession(true);
session.setAttribute("data", data);
```

JSP

<pre>&lt;c:forEach items="\${sessionScope.data}" var="s"&gt;     \${s.rollnum} \${s.name}&lt;br/&gt; &lt;/c:forEach&gt;</pre>	<pre>\${sessionScope["data"]}</pre>
---	-------------------------------------

### Example 3: <c:if>

```
<form method="post">
    Enter a salary:
    <input type="text" name="salary"/><br/>
    <c:set var="salary" scope="request" value="${param.salary}"/>
    <c:if test="${salary<200}">
        <h3 style="color: blue">Tax: ${0}</h3>
    </c:if>
    <c:if test="${(salary>=200)&&(salary<400)}">
        <h3 style="color: blue">Tax: ${(salary-200)*0.1}</h3>
    </c:if>
    <c:if test="${salary>=400}">
        <h3 style="color: blue">Tax: ${(200*0.1)+((salary-400)*0.15)}</h3>
    </c:if>
    <input type="submit" value="Calculate tax"/>
</form>
```

### Example 4: <c:choose>

```
<c:choose>
    <c:when test="${salary<200}">
        <h3 style="color: blue">Tax: ${0}</h3>
    </c:when>
    <c:when test="${(salary>=200)&&(salary<400)}">
        <h3 style="color: blue">Tax: ${(salary-200)*0.1}</h3>
    </c:when>
    <c:when test="${salary>=400}">
        <h3 style="color: blue">Tax: ${(200*0.1)+((salary-400)*0.15)}</h3>
    </c:when>
    <c:otherwise>
        salary is undetermined...
    </c:otherwise>
</c:choose>
```

## JSTL **Formatting** Tags

```
<fmt:formatNumber pattern="###.###" value="${Math.PI}"/>

<fmt:formatDate pattern="dd/MM/yyyy" value="%=new Date()%" />
```

## JSTL Function Tags

The JSTL function provides a number of standard functions, most of these functions are common string manipulation functions. The syntax used for including JSTL function library in your JSP is:

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

### Example

```
<form method="post">
    Enter a paragraph:
    <input type="text" name="txt"/><br/>
    <c:set var="ftxt" value="${param.txt}" scope="request"/>
    <h3 style="color: blue">
        the number: ${fn:length(ftxt)}</h3>
    <br/>
    <input type="submit" value="Submit"/>
</form>
```