**Lý thuyết:**

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

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

Mã JSP (Scriplets):

<% code %>

Ví dụ viết vào văn bản:

<% out.println(“Welcome to JSP world”); %>

<% out.write("How are you?\n");%>

Khai báo (Declarations):

<%! %>

Biểu thức (Expressions):

<%= expression %>

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

<%! String name="Tô Lan Hương"; %>

<!-- đây là chú thích -->

<p>chú thích : <!-- <%= name %> --></p>

<%--hidden comment --%>

<%// hidden comment %>

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

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

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

|  |  |
| --- | --- |
| **Directive** | **Miêu tả** |
| <%@ page ... %> | Định nghĩa một thuộc tính page-dependent (phụ thuộc trang), như ngôn ngữ scripting, trang lỗi và các yêu cầu bộ đệm |
| <%@ include ... %> | Include một file trong suốt giai đoạn phiên dịch |
| <%@ taglib ... %> | Khai báo một thư viện thẻ, chứa các action tùy biến, được sử dụng trong trang đó |

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/xml" %>

<%@ page contentType="text/html" %>

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

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

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

1. **include** Directive trong JSP được sử dụng để bao một file trong suốt giai đoạn phiên 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="content.jsp" %>

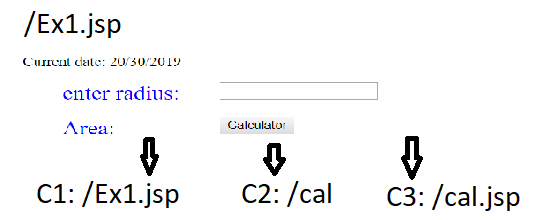
1. Java Server Pages API cho phép định nghĩa các Custom Tags trong JSP, mà giống như các thẻ HTML hoặc XML và một thư việc thẻ (tag library) là một tập hợp các thẻ được định nghĩa bởi người sử dụng mà triển khai các hành vi tùy biến.

**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.

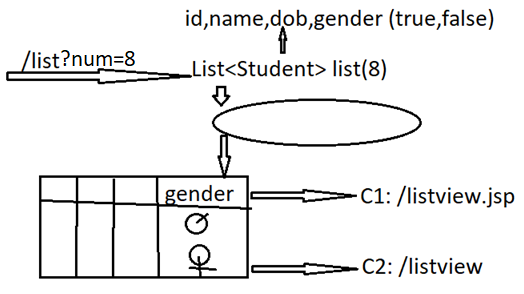
<%@ taglib uri="uri" prefix="prefixOfTag" >

JSPDemo.jsp

**Bài 1:**

****

**Bài 2:**

****

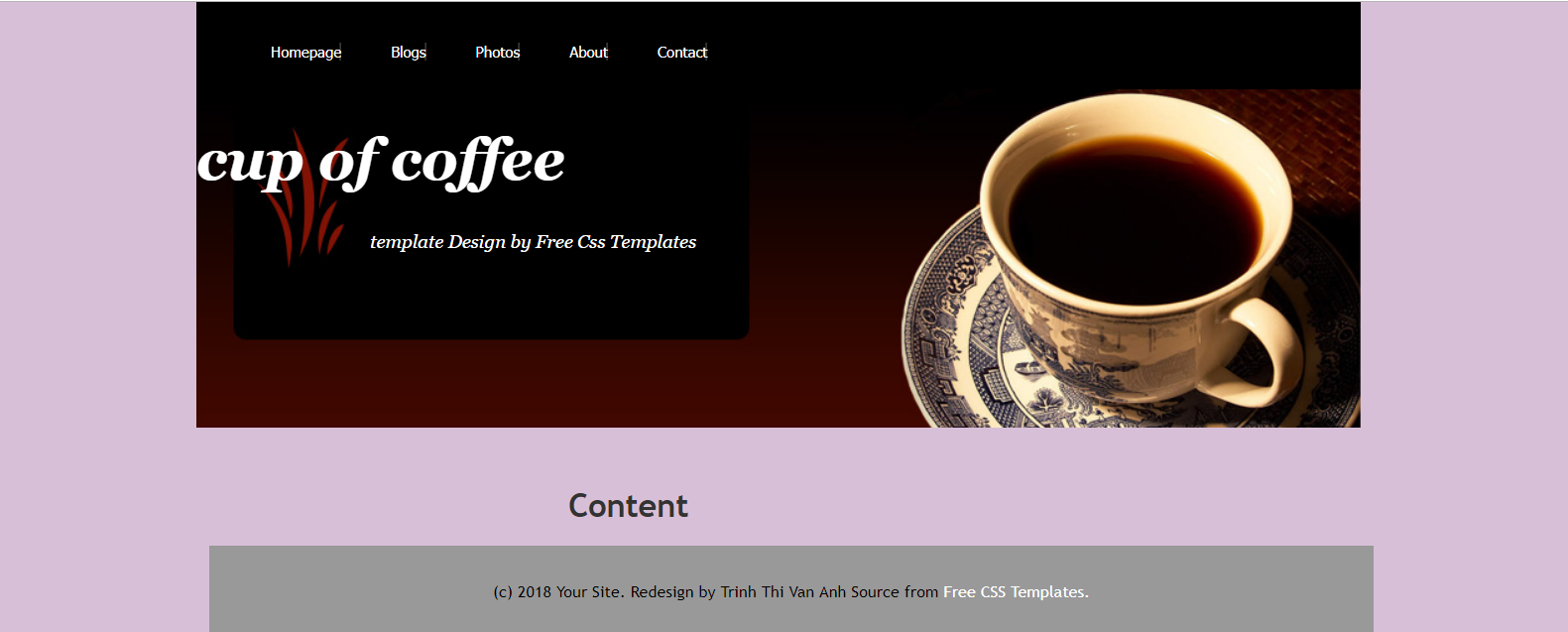
**Bài 3:**

Tạo trang gồm nhiều file

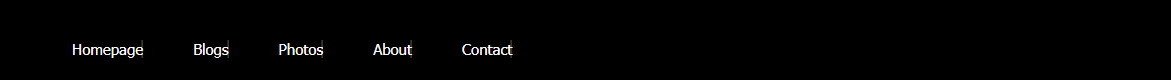
Ví dụ:

Index.jsp:

|  |
| --- |
| <div id="wrapper">  <%@include file="top.jsp" %>    <%@include file="logo.jsp" %>  <div id="content"><h1>Content</h1>    </div>  <%@include file="footer.jsp" %>  </div> |



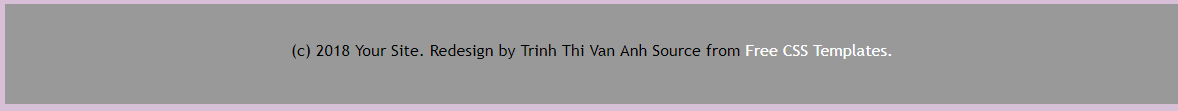
Top.jsp



Logo.jsp



Footer.jsp



**Lý thuyết**

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" />

Ví dụ: có file date.jsp

|  |
| --- |
| <p>  Today's date: <%= (new java.util.Date()).toLocaleString()%>  </p> |

File index.jsp:

|  |
| --- |
| <h2>The include action Example</h2>  <jsp:include page="date.jsp" flush="true" /> |

Bai tap de lam:

<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”

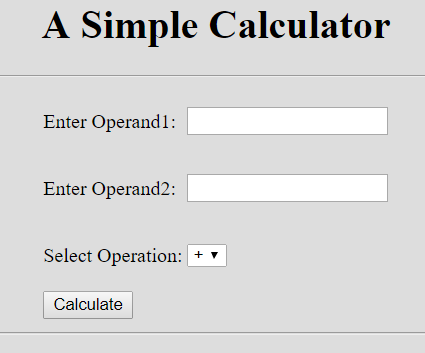
|  |
| --- |
| <jsp:useBean id="students" class = "bean.StudentsBean">  Tương đương  <% bean.StudentsBean students = new bean.StudentsBean(); %> |

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="Ha"/>  Tương đương <%= students.setFirstName(“Ha”) %>  <jsp:setProperty name="students" property="**\***"> |

**Bài 4:**

Làm lại bài tính số học dùng java Bean (JavaBeanEx)



B1: java bean (lop – java)

B2: jsp (form) – ket qua

**Bài 5:**

1. Cho Project: Ex-DrawCircle với các file có sẵn. file index.html khi chạy được

Tạo servlet có tên là CircleServlet và url-pattern /circle, servlet sinh ngẫu nhiên ra các đường tròn lưu trữ trong ServletScope với tên “cirs” sao đó được hiển thị bởi /circleview

1. Cho Project: Ex-DrawRect với các file có sẵn. file index.html khi chạy được



Tạo servlet có tên là RectServlet và url-pattern /rect, servlet sinh ngẫu nhiên ra các đường chữ nhật lưu trữ trong ServletScope với tên “rects” sao đó được hiển thị bởi /rect.jsp