**Bài 10 JSP & JSTL**

Module: JWBD

# Kiểm tra bàitrước

Hỏivàtraođổivềcáckhókhăngặpphảitrongbài“Tổngquanứngdung Web"

Tómtắtlạicácphầnđãhọctừbài“CâulệnhtruyvấnSQL”

## Mụctiêu

* Sửdụng đượcJSP
* SửdụngđượcformtrongứngdụngwebdựatrênServlet&JSP
* Sửdụngđượccácthẻjstlthôngdụng

**Java ServerPage**

Tìm hiểu vềJSP

Luồng hoạtđộngcủamộtứngdụngwebsửdụngJSP

Vòng đờiJSP

**JSP làgì?**

* JSP (JavaServer Pages) là một ngôn ngữ kịch bản phía server, cho phépngườidùngtạoracáctrangwebđộng.
* JSPđượcpháthànhvàonăm1999bởiSunMicrosystems,đượcchạy trênnềnJDK1.3trởvềsau,làmộtcôngnghệkhôngthểthiếucủa JavaEE.PhiênbảnmớinhấtcủaJSPlà2.2.
* JSPlà một tài liệu text có thể trả về cả static content (như HTML, XML,text)vàdynamiccontent(nhưmãnguồnjava,cácthuộctính của các lớp Java Bean, các custom tag) cho trình duyệt. Static contentvàdynamiccontentcóthểđanxenlẫnnhau.
* CácfileJSPđượclưuvớiphầnmởrộnglà.jsp.
* Khônggiống nhưServlet,cácfile JSPsẽtựđộngbiêndịchvàtriển khaikhicóbấtkỳthayđổinào.

## Ví dụ fileindex.jsp



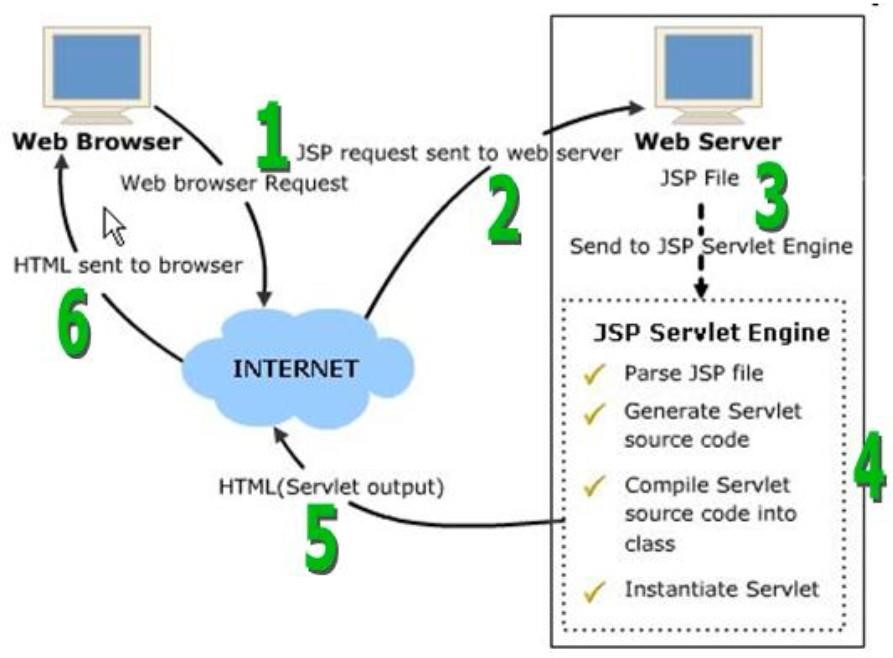
## So sánh JSP vàServlet

* Vềbảnchất,JSPcũngchínhlàServlet.Vìtrongquátrìnhbiêndịch, JSPsẽđượcchuyểnthànhServletrồiServletnàysẽđượcbiêndịch. Servlet của trang JSP có vòng đời giống như các Servlet thông thường.
* **Giốngnhau**
* Đềunằmởphíaserver.
* Đềuxửlýdữliệu“động”.
* Đềuchạyvớiwebcontainer.
* ĐềulànhữngcôngnghệquantrọngcủaJavaEE.

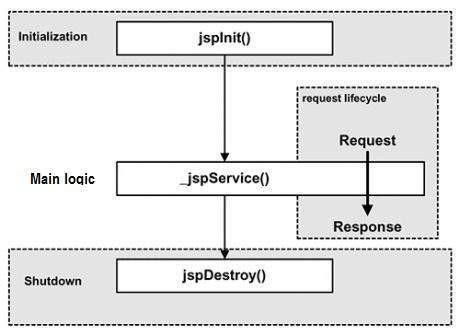
## So sánh JSP vàServlet

* **Khácnhau**
* Servletrấtmạnhvềxửlývàđiềuphối,nhưnglạirấtyếuvềtạogiaodiệnvà bảo trìweb.
* JSPmạnhvềxửlýhiểnthịnhưnglạiyếuvềxửlýnghiệpvụvàđiềuphối.
* Trong thực tế, chúng ta kết hợp sức mạnh của Servlet và JSP vào mô hình MVCthì:ServletđóngvaitròController,thìJSPđóngvaitròView.
* Ở Servlet, mã HTML nằm trong mã Java. Còn ở JSP, mã Java nằm trong mã HTML.

## Cách xử lý trangJSP



## Vòng đời của một trangJSP

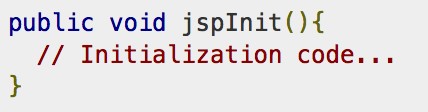
* Vòng đời của một JSP được tính từ khi JSP đó được tạo ra cho đến khi bịhủy bỏ.
* CácgiaiđoạntrongvòngđờitrangJSP
* Biêndịch
* Khởitạo
* Thựcthi
* Huỷ

## Biêndịch

* Khitrìnhduyệtyêucầu1trangJSP,JSPengineđầutiênsẽkiểmtra xemcó cầnbiêndịch trangđó không.NếutrangJSPchưabaogiờ đượcbiêndịchhoặcđãđượcchỉnhsửakểtừlầnbiêndịchcuốicùng thìJSPenginesẽbiêndịchtrangJSP.
* Quátrìnhbiêndịchgồm3bước:
* Phân tíchtrang JSP
* Chuyển trang JSP sangServlet
* Biên dịchServlet

## Khởitạo

* Khi JSP container nạp nạp trang JSP, nó sẽ gọi phương thứcjspInit()trướckhitrảlờicácrequestkhác.
* Nếu bạn cần thực hiện sự khởi tạo JSP riêng, ghi đề phương thức jspInit():



* Việckhởitạođượcthựchiệnchỉmộtlầnvàvớiphươngthứcintcủa

Servlet

* TrongjspInit(),chúngtacóthểkhởitạokếtnốitớiCSDL,mởfile,…

## Thựcthi

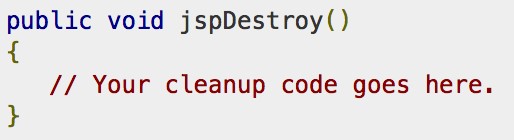
* KhitrìnhduyệtyêucầumộttrangJSP,trangnàyđã đượcnạpvàkhởitạo, thìJSPenginesẽgọiphươngthức\_jspService().
* Phương thức \_jspService() nhận một HttpServletRequest và một HttpServletResponsenhưlàcácthamsốcủanó.
* Phương thức \_jspService() của JSP được triệu hồi một lần cho mỗi yêu cầuvànóchịutráchnhiệmtạoResponsechoRequestđó.



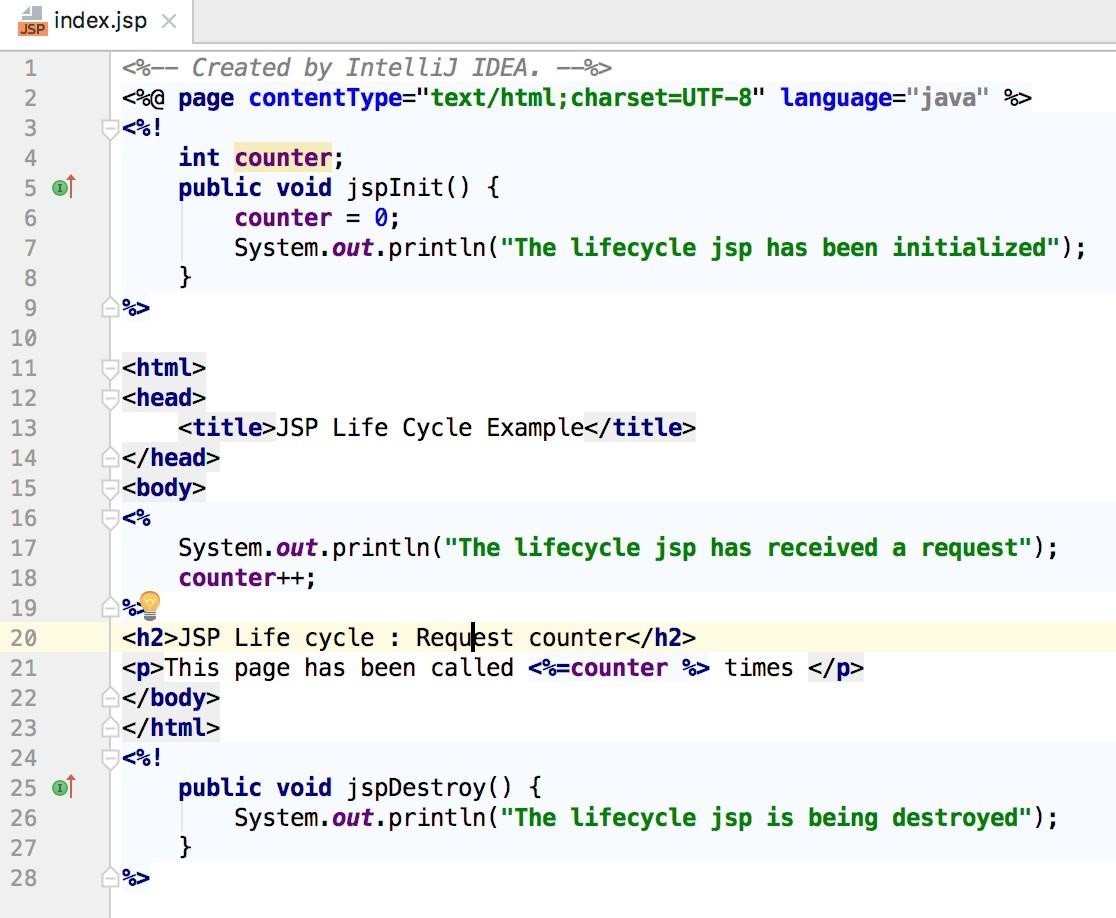
* Phương thức này cũng chịu trách nhiệm tạo các phản hồi tới tất cả phươngthứccủaHTTP,vídụ:GET,POST,DELETE,…

## 4.Huỷ

* GiaiđoạnhủymộtJSPtrongvòngđời JSPbiểuthịkhinàothìmột JSPbịgỡbỏkhỏimộtcontainer.
* Ghi đè phương thức jspDestroy khi bạn cần thực hiện bất kỳ quá trìnhhủynào,vídụnhưgiảiphóngkếtnốivớiDatabase,hoặcđóng cácfile.



## Vídụ



## Cú pháp cơ bản trongJSP

* ĐểchèncácđoạnmãnguồnjavavàobêntrongtrangJSPsửdụng3 dạng cơbản sau:
* Expressions: <%= Expressions%>
* Scriptlets: <% Code%>
* Declarations: <%!Declarations%>

## Các đối tượngẩn

* Đối tượng ẩn (Implicit Object) là các đối tượng Java mà JSP Containercungcấpchocácnhàpháttriểntrongmỗitrang.
* Các đối tượng này được gọi trực tiếp mà không cần khai báo. Còn đượcgọilàcácbiếnđượcđịnhnghĩatrước.

## Các đối tượngẩn

|  |  |
| --- | --- |
| **Đối tượng** | **Mô tả** |
| **request** | Đây là đối tượng HttpServletRequest được liên kết với request |
| **response** | Đây là đối tượng HttpServletResponse được liên kết với response tới client |
| **out** | Đây là đối tượng PrintWriter được sử dụng để gửi dữ liệu tới client |
| **session** | Đây là đối tượng HttpSession được liên kết với request |
| **application** | Đây là đối tượng ServletContext được liên kết với applicationcontext |
| **config** | Đây là đối tượng **ServletConfig** được liên kết vớipage. |
| **pageContext** | Điều này đóng gói việc sử dụng các tính năng cụ thể của server như **JspWriters** với hiệu suất cao hơn. |
| **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 biêndịch. |
| **exception** | Đối tượng **Exception** cho phép các dữ liệu ngoại lệ được truy cập bằng JSP được chỉ định. |

## JSTL

* JSTLlàbộthưviệnthẻchuẩnđượcbổsungvớimụcdicdhstốiưu lập trình**trong** JSP
* CácthưviệncầnthiếtchoJSTLgồm
* Jstl-api.jar
* Jstl-impl.jar
* TrongJSTLcórấtnhiềubộthẻđểxửlýcácvấnđềkhácnhau
* Core:Chứacácthẻđiềukhiểncơbản
* Format:Chứacácthẻđịnhdạngvàđangônngữ
* XML:Chứacácthẻxửlýtàiliệuxml
* SQL:ChứacácthẻlàmviệcvớiCSDL
* Function:ChứacácthẻcungcấpcáchàmhỗtrợchoExpressionLanguage **Trongphạmvibàihọcnày,chúngtảsẽsửdụngbộthẻcơbản**

## Các thẻ cơ bản trongjslt

* Cácthẻcơbản(CoreTags)
* Chứa cać lệnh điều khiển như như if, choose when otherwise,vòng lặp nhưforEach….ĐểsửdụngJSTLcoretags,bạncầnkhaibáovàotrongtrangJSPđangsửdụngnhưsau:

<%@tagliburi=[”http://java.sun.com/jsp/jstl/core”](http://java.sun.com/jsp/jstl/core)prefix=”c”%>

## Các thẻ JSTL cơbản

|  |  |
| --- | --- |
| **Thẻ** | **Môtả** |
| **Thẻ <c:out > trongJSTL** | **Giống <%= ... >, nhưng cho cácExpression** |
| **Thẻ <c:set > trongJSTL** | **ThiếtlậpkếtquảcủamộtướclượngExpressiontrongmột'scope'** |
| **Thẻ <c:remove > trongJSTL** | **Gỡ bỏ một biến mục tiêu (từ một biến scope cụ thể, nếu đãxác định)** |
| **Thẻ <c:catch> trongJSTL** | **BắtbấtkỳThrowablemàxuấthiệntrongthâncủanóvàtrưngbàynó một cách tùyý** |
| **Thẻ <c:if> trongJSTL** | **Thẻđiềukiệnđơngiản,màướclượngphầnthâncủa nónếuđiềukiện đã cho làtrue** |
| **Thẻ <c:choose> trongJSTL** | **Thẻ điều kiện đơn giản mà thiết lập một context cho các hoạt động điều kiện loại trừ, được đánh dấubởi <when> và <otherwise>** |
| **Thẻ <c:when> trongJSTL** | **Thẻ phụ của <choose> mà include phần thân của nó nếu điềukiện được ước lượng làtrue** |

## Các thẻ JSTL cơbản

|  |  |
| --- | --- |
| **Thẻ** | **Môtả** |
| **Thẻ <c:otherwise > trongJSTL** | **Thẻ phụ của <choose> mà theo sau thẻ <when> và chỉ chạy nếu tất cả điềukiệntrướcđượcướclượnglà'false'** |
| **Thẻ <c:import> trongJSTL** | **ThunhậnmộtURLtuyệtđốihoặcquanhệvàtrưngbàynộidungcủa nótớihoặctrangđó,mộtStringtrong'var',hoặcmộtReadertrong 'varReader'** |
| **Thẻ <c:forEach > t~~r~~ongJSTL** | **Thẻ lặp cơ bản, chấp nhận nhiều kiểu tập hợp khác nhau và hỗ trợ subsetting(chia tập con) và tính năngkhác** |
| **Thẻ <c:forTokens> trongJSTL** | **Lặp qua các token, được phân biệt bởi các dấu phân tách (delimiter) đã cungcấp** |
| **Thẻ <c:param> trongJSTL** | **Thêm một parameter tới một URL của thẻ đang chứa'import'** |
| **Thẻ <c:redirect > trongJSTL** | **Redirect tới một URLmới** |
| **Thẻ <c:url> trongJSTL** | **Tạo một URL với các tham số truy vấn tùyý** |

## dụngc:out

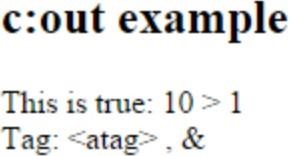
<%@ **taglib uri**="[**http://java.sun.com/jsp/jstl/co**](http://java.sun.com/jsp/jstl/c)**re**" **prefix**="**c**" %>

<!DOCTYPE **html**>

<**html**>

<**head**>

<**meta charset="UTF-8"**>

<**title**>c:out example</**title**>

</**head**>

<**body**>

<**h2**>c:out example</**h2**>

<**c:out value="${'This is true: 10 > 1 '}"** />

<**br**/>

Tag: <**c:out value="${'<atag> , &'}"**/>

</**body**>

</**html**>

## dụngc:set

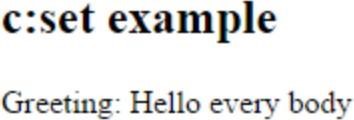
<%@ **taglib uri**="[**http://java.sun.com/jsp/jstl/co**](http://java.sun.com/jsp/jstl/c)**re**"**prefix**="**c**"

%>

<!DOCTYPE **html**>

<**html**>

<**head**>

<**meta charset="UTF-8"**>

<**title**>c:set example</**title**>

</**head**>

<**body**>

<**h2**>c:set example</**h2**>

<**c:set scope="request" var="greeting" value="Hello every body"** />

Greeting: <**c:out value="${**greeting**}"**/>

</**body**>

</**html**>

## dụngc:if

<%@ **taglib uri** = ["**http://java.sun.com/jsp/jstl/co**](http://java.sun.com/jsp/jstl/c)**re**" **prefix** = "**c**" %>

<**html**>

<**head**>

<**title**><**c:if**> Tag Example</**title**>

</**head**>

<**body**>

<**c:set var = "salary" scope = "session" value = "${**2000\*2**}"**/>

<**c:if test = "${**salary > 2000**}"**>

<**p**>My salary is: <**c:out value = "${**salary**}"**/><**p**> </**c:if**>

</**body**>

</**html**> 

## dụng c:choose -c:when -c:otherwise

<%@ **taglib uri** = ["**http://java.sun.com/jsp/jstl/core**](http://java.sun.com/jsp/jstl/core)" **prefix** = "**c**"%>

<**html**>

<**head**>

<**title**><**c:choose**> TagExample</**title**>

</**head**>

<**body**>

<**c:set var = "salary" scope= "session" value = "${**2000\*2**}"**/>

<**p**>Your salary is : <**c:out value ="${**salary**}"**/></**p**>

<**c:choose**>

<**c:when test = "${**salary <= 0**}"**> Salary is very low to survive.

</**c:when**>

<**c:when test = "${**salary > 1000**}"**> Salary is very good.

</**c:when**> <**c:otherwise**> No comment sir...

</**c:otherwise**>

</**c:choose**>

</**body**>

</**html**>

## c:forEach

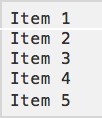
<%@ **taglib uri** = ["**http://java.sun.com/jsp/jstl/co**](http://java.sun.com/jsp/jstl/c)**re**" **prefix** =

"**c**"%>

<**html**>

<**head**>

<**title**><**c:forEach**> TagExample</**title**>

</**head**>

<**body**>

<**c:forEach var = "i" begin = "1" end = "5"**> Item

<**c:out value = "${**i**}"**/><**p**>

</**c:forEach**>

</**body**>

</**html**>

## Tóm tắt bàihọc

* JSP (JavaServer Pages) là một ngôn ngữ kịch bản phía server, cho phépngườidùngtạoracáctrangwebđộng.
* ToànbộtrangJSPđượcthôngdịchsangServlet(mộtlần)vàServlet đượcthựcthikhiyêucầucủaclientgửiđến.

# Hướngdẫn

Hướngdẫnlàmbàithựchànhvàbàitập

Chuẩnbịbàitiếptheo:MVC