

JSP 期末報告

學號：110916040

姓名：游博丞

一、 投影片補充



- Java 用於開發網頁應用系統的技術
 - 基於 Java Platform Enterprise Edition (Java EE)
 - 目前版本 JSP 2.3
- 採用範本系統
 - HTML 中嵌入 Java

2

Java Platform Enterprise Edition (Java EE)是由一系列的技術標準所組成的平台，包括：

- Applet – Java Applet
- EJB - 企業級 JavaBean(Enterprise Java Bean)
- JAAS - Java Authentication and Authorization Service
- JACC – J2EE Authentication Contract for Containers
- JAF – Java Beans Activation Framework
- JAX-RPC – Java API for XML-Based Remote Procedure Calls

- JAX-WS – Java API for XML Web Services
- JAXM – Java API for XML Messaging
- JAXP – Java XML 解析 API(Java API for XML Processing)
- JAXR – Java API for XML Registries
- JCA – J2EE 連接器架構(J2EE Connector Architecture)
- JDBC – Java 數據庫連接(Java Database Connectivity)
- JMS – Java 消息服務(Java Message Service)
- JMX – Java Management
- JNDI – Java 名稱與目錄接口(Java Naming and Directory Interface)
- JSF – Java Server Faces
- JSP – Java 服務器頁面(Java Server Pages)
- JSTL – Java 服務器頁面標準標籤庫(Java Server Pages Standard Tag Library)
- JTA – Java 事務 API(Java Transaction API)
- JavaMail
- Servlet – Java Servlet API
- StAX – Streaming APIs for XML Parsers
- WS – Web Services

資料來源：[Jakarta EE wiki](#)



Resin

- Caucho 開發
 - 提供 Java 解決方案的公司
- Java EE application server
 - Java Container
 - Web Server

4

Resin 雖然是基於 Java 的服務器，但其核心是用高度優化的 C 語言進行編寫。

Caucho 表示 Java 是允許 Resin “full featured” 的 layer，而 C 則是提供了速度。

資料來源：[Resin \(software\)](#)



註解 in JSP

- ch3/comments.jsp

```
<%-- JSP註解 --%>
<HTML>
<HEAD><TITLE>Hello World JSP!</TITLE></HEAD>
<BODY>
<!-- HTML註解 -->
<%
  out.println("<FONT COLOR = RED >Hello World!</FONT><BR>");
  // Java 註解
%>
</BODY>
</HTML>
```

5

<!--HTML 註解

<%-- JSP 註解

//Java 註解(單行註解)

/*多行

註解*/

--%>

-->

資料來源：[註解元素](#)

Scriptlets

- JSP 中有關主程式部分
- 語法如下

```
<%  
    scriptlets  
%>
```
- Examples
 - `<% i = 1; %>`
 - `<% i++; %>`

語法與
Java 相同

10

直接在 JSP 中撰寫的 HTML，都會變成 out 物件所輸出的內容。

Expressions

- 簡化 JSP 輸出程式撰寫
- 語法如下

```
<%= expression %>
```
- Examples
 - `<%= i %>`
 - `<%= 2*2 %>`

僅能單行，
不能換行，
而且結尾不能有";"

11

運算式元素中的運算式，會直接轉譯成 out 物件輸出時的指定內容，這也是運算式元素中不用加上分號的原因。

資料來源：[宣告、Scriptlet 與運算式元素](#)

jspInit()

- JSP 第一次被瀏覽時執行，用於資料初始化
- JSP 再次被瀏覽時，不會再執行
- 定義在 Declarations 中，語法如下

```
public void jspInit() {...}
```

22

若想要在 JSP 網頁載入執行時執行某些初始動作，可以重新定義 jspInit() 方法。

資料來源：[JSP 生命週期](#)



page directive

- 語法如下

```
<%@ page attribute1="value1"
attribute2="value2" ... %>
```
- 常見 attributes
 - language
 - 定義 JSP Page 使用的描述語言，但目前只有 Java
 - 語法如下

```
<%@ page language="描述語言" %>
```

29

常見 attributes：

- pageEncoding

告知容器此 JSP 網頁中的文字編碼，若網頁中包括非 ASCII 編碼範圍中的字元，需指定正確的編碼格式，避免出現亂碼。

- isELIgnored

設定 JSP 網頁中是否有忽略運算式語言(Expression Language)，預設是 false，如果設定為 true，則不會轉譯運算式語言。此設定會覆蓋 web.xml 中的<el-ignored>設定。

include directive

- 在 Translation Phase 將指定檔案內容插入產生的 .java 檔
 - 該檔案內容可包含 HTML 及 JSP 程式
- 語法如下
 - `<%@ include file="檔案名稱" %>`
- Examples
 - `<%@ include file="heading.inc" %>`
 - `pageInclude.jsp`pageInclude.jsp ;
`heading.inc`

48

Include 多個檔案時，只會產生一個 Servlet。使用指令元素 include 來包括其他網頁的內容時，會再轉譯時期就決定了轉譯後的 Servlet 內容，式一種靜態的指定方式。

資料來源：[指示元素](#)

Actions (cont.)

- `<jsp:useBean>`
 - 宣告 JSP Page 使用的 JavaBean 物件
- `<jsp:setProperty>`
 - 設定所使用 JavaBean 物件的屬性
- `<jsp:getProperty>`
 - 取得所使用 JavaBean 物件的屬性

57

`<jsp:useBean>`

宣告 JSP Page 使用的 JavaBean 物件，此處所指的 JavaBean 並非桌面系統或 EJB(Enterprise JavaBeans)中的 JavaBean 物件，而是滿足以下條件的純粹 Java 物件：

- 必須實作 `java.io.Serializable` 介面
- 沒有公開(`public`)的類別變數
- 具有無參數的建構式
- 具有公開的設值方法(`Setter`)與取值方法(`Getter`)

其 `id` 屬性用於指定 `JavaBean` 實例的參考名稱，後面在使用

`<jsp:setProperty>` 或 `<jsp:getProperty>` 標籤時，可以依據這個名稱來取得所建立的 `JavaBean` 名稱。`class` 屬性用以指定實例化哪一個類別。`scope` 指定可先找哪個屬性範圍是否有 `JavaBean` 的屬性存在。

由於 `JSP` 網頁最終將轉換為 `Servlet`，所謂的 `JavaBean` 其實也是 `Servlet` 的一個物件實例，當使用 `<jsp:useBean>` 時，實際上就是在宣告一個 `JavaBean` 的物件，`id` 屬性用以指定參考名稱，`class` 屬性則是型態名稱。

使用 `<jsp:useBean>` 標籤時，會在屬性範圍(預設為 `page`)中尋找是否有 `id` 名稱所指定的屬性，若有就直接使用，無則建立新物件。

屬性範圍可以用 `scope` 屬性來指定其儲存的屬性範圍，可指定的值有 `page`(預設)、`request`、`session`、`application`。

若使用 `<jsp:useBean>` 標籤時沒有指定 `scope`，則預設只在 `page` 範圍中尋找 `JavaBean`，找不到就直接建立新的 `JavaBean` 物件，不會再到 `request`、`session`、`application` 中尋找。

若是想在轉譯後的 Servlet 程式碼中指定宣告 JavaBean 時的型態，可使用 type 屬性。

type 屬性的設定可以是一個抽象類別，也可以是一個介面。如果只有設定 type 而沒有設定 class 屬性，則必須確定在某個屬性範圍中已存在所要的物件，不然會發生 InstantiationException 例外。

<jsp:setProperty>

設定所使用 JavaBean 物件的屬性，name 屬性用於指定要使用哪個名稱取得 JavaBean 實例，property 屬性設定“*”表示將自動尋找符合 JavaBean 中設值方法名稱的請求參數值，若請求參數名稱為 xxx，就將請求參數值使用 setXxx()方法設定給 JavaBean 實例。

若請求參數中包括 password，則會透過 JavaBean 的 setPassword()方法設定給 JavaBean 實例。也可以不指定請求參數名稱，而由 JSP 的自省 (Introspection)機制來判斷是否有相同的請求參數名稱，若有的話就自動找出對應的設值方法並呼叫以設值給 JavaBean。

若 JavaBean 屬性為整數或浮點數之類的基本型態，自省機制可以自動轉換請求參數字串為對應屬性的基本資料型態。

<jsp:getProperty>

取得所使用 JavaBean 物件的屬性，name 屬性用於指定要使用哪個名稱取得 JavaBean 實例，property 屬性則指定要取得哪個屬性值，若指定為 xxx，則使用 getXxx()方法取得 JavaBean 屬性值並顯示在網頁上。

資料來源：

1. [簡介 <jsp:useBean>、<jsp:setProperty> 與 <jsp:getProperty>](#)
2. [深入 <jsp:useBean>、<jsp:setProperty> 與 <jsp:getProperty>](#)



- get：通常用在較小或非敏感的資料，因為資料會被放在網址中直接傳出，容易被看到資料。

- post：post 會將表單資料放在 HTTP 傳輸封包 body 中送出。通常被用在表單資料量比較大、有夾帶檔案上傳(file upload)或隱私性考量的資料。

資料來源：[HTML 表單 \(form\)](#)



Cookie

- 儲存於瀏覽器，供瀏覽器與 Web Server 間交換資料用的檔案
- Cookie 變數儲存於 Cookie 中，供網頁程式於 Web Server 執行時存取
- 使用 session 物件時通常也需要 Cookie

21

cookie 作用：

- 對特定物件的追蹤
- 儲存使用者網頁瀏覽紀錄與習慣
- 簡化登入

安全風險：容易洩漏使用者資訊。

資料來源：[JSP 學習筆記七之 Cookie](#)



使 session object 無效

- 可清除所有儲存資料，常用於模擬使用者連線登出
- 語法如下
`void session.invalidate()`

10

Session 物件無效的三種情況：

- 呼叫了 `session.invalidate()`;
- session 過期
- 伺服器重啟

資料來源：[JSP 學習筆記四之 out 和 session 物件](#)

application object (cont.)

- 常見功能
 - 屬性存取
 - 取得版本資訊
 - 將訊息寫至記錄檔(log file)
 - 取得 Web 應用程式的初始參數
 - 取得伺服器的資訊或相關資源
- 詳細參見 [javax.servlet.ServletContext](#) 的說明

30

- 實現使用者之間資料的共享，可存放全域變數。
- 在使用者的前後連線或不同使用者之間的連線中，可對 application 物件的同一屬性進行操作。
- 在任何地方對 application 物件屬性的操作，都將影響其他使用者對此的訪問。

資料來源：[JSP 學習筆記五之 application 和 page 物件](#)

二、 額外補充

運算式語言(Expression Language/EL)：

- EL 為 Java 中的一種通用編程語言，主要作用在 Java Web 應用程式嵌入到網頁(JSP)中，用以訪問頁面上下文及不同作用域中的對象，取值或進行簡單的運算/判斷操作。
- EL 是使用 `{ }` 來包括要進行處理之運算式，可使用點運算子 `.` 指定要存取的屬性，使用加號 `+` 運算子進行加法運算。
- `param` 是 EL 隱含物件之一，表示使用者的請求參數，`param.x` 表示取得使用者所發出的請求參數 `x` 的值。
- EL 的點運算子可以像 Java 程式碼一般連續存取物件。
- EL 對於 `null` 值會直接以空字串顯示，而不是直接顯示 `null` 值，在進行運算時，也不會因此發生錯誤而拋出例外。
- `pageContext` 也是 EL 的隱含物件之一，透過點運算子接上 `xxx` 名稱，表示呼叫 `getXxx()` 方法，如需轉換型態，EL 也會自行處理
- 要判斷 JSP 網頁是否使用 EL，可使用 `page` 指示元素的 `isELIgnored` 屬性 (預設為 `false`) 進行判斷。
- 若 JSP 網頁已使用 `page` 指令元素的 `isELIgnored` 設定是否支援 EL，則不用理會 `web.xml` 中的 `<el-ignored>` 的設定為何。

資料來源：

1. [表達式語言 wiki](#)
2. [簡介 EL](#)