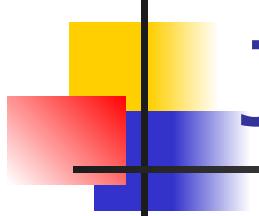


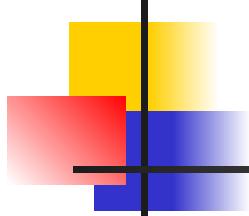
JSP 基本語法

國立台北教育大學
資訊科學系
游象甫



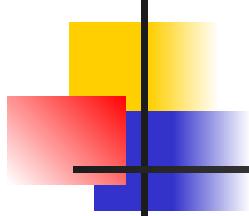
JSP Page

- Elements
 - Known by JSP container
- Template data
 - Not known by JSP container, such as HTML



Elements

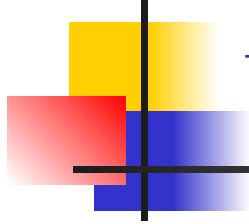
- Directives
 - Provide global information independent of any specific request received by the JSP
 - Provide information for the translation phase
- Actions
 - Provide information for the request processing phase



Elements (cont.)

■ Scripting Elements

- Provide "glue" around template text and actions
- Manipulate objects and perform computation that affects generated results



註解 in JSP

■ ch3/comments.jsp

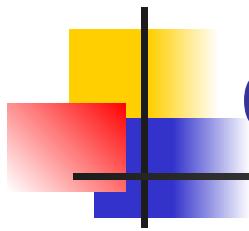
```
<%-- JSP註解 --%>
<HTML>
<HEAD><TITLE>Hello World JSP!</TITLE></HEAD>
<BODY>
<!-- HTML註解 -->
<%
  out.println("<FONT COLOR = RED >Hello World!</FONT><BR>");
  // Java 註解
%>
</BODY>
</HTML>
```

ch3/comments.jsp

■ _comments_jsp.java



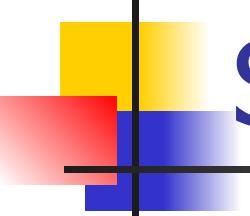
ch3/comments.jsp (cont.)



A screenshot of a Firefox browser window showing the source code of a JSP file. The title bar shows two tabs: "Hello World JSP!" and "view-source:127.0.0.1:8080/ch3/comments.jsp". The main content area displays the following Java Server Page code:

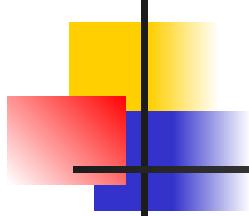
```
1 <HTML>
2 <HEAD><TITLE>Hello World JSP!</TITLE></HEAD>
3 <BODY>
4 <!-- HTML -->
5 <FONT COLOR = RED >Hello World!</FONT><BR>
6
7
8 </BODY>
9 </HTML>
```

The browser's status bar at the bottom shows the date and time: "下午 10:44 2021/3/12".



Scripting Elements

- JSP 中除 HTML 外，主要的程式碼，負責動態輸出結果
- 三個類別如下
 - Declarations
 - Scriptlets
 - Expressions



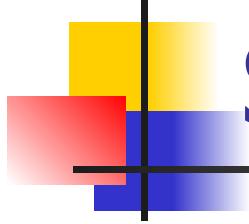
Declarations

- 宣告變數與函數(或 method), 成為 JSP 對應 Servlet 的 data field 與 method
- 語法如下

```
<%!
    declarations
%>
```

宣告語法
與
Java 相同

- Examples
 - <%! int i; %>
 - <%! int i = 0; %>



Scriptlets

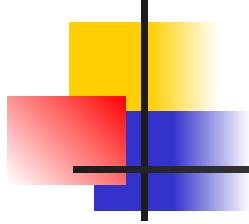
- JSP 中有關主程式部分
- 語法如下

<%
 scriptlets
%>

語法與
Java 相同

- Examples

- <% i = 1; %>
- <% i++; %>



Expressions

- 簡化 JSP 輸出程式撰寫
- 語法如下
`<%= expression %>` *↑* `out.print("expression");`
- Examples
 - `<%= i %>`
 - `<%= 2*2 %>`

僅能單行，
不能換行，
而且結尾不能有";
"

JSP Examples

■ ch3/array.jsp

```
<%!
// Declarations
int i;
int a[];
%>
<FONT SIZE = 5 COLOR = BLUE>變數的使用</FONT>
<HR>
<%
// Scriptlets
a=new int[3];
a[0]=9;
a[1]=6;

for(i=0; i<a.length; i++) {
    out.print("a["+i+"]=" + a[i] + ",");
%><%-- Expressions --%>
    <%= a[i] %>&nbsp //html 表示空格
    // &nbsp 可以用out.print("&nbsp") 取代
<%
}
out.println("<br>Length of array = " + a.length + ",");
%>
<%= a.length %> → expression
```

HTML 語法

Scriptlet

ch3/array.jsp

■ _array_jsp.java



變數的使用

```
a[0]=9, 9 a[1]=6, 6 a[2]=0, 0  
Length of array = 3, 3
```

ch3/count.jsp

Servlet (instance)

↓ ↓ ... ↓
thread thread thread

```
<%!
    // Declarations
    int count = 0;
%>
<FONT SIZE = 5 COLOR = BLUE>變數的使用</FONT>
<HR>
<%
    // Scriptlets
    int lcount = 0;

    count++;
    lcount++;
%>
```

) instance

) threads

累積瀏覽人數 <%= count %>, 本次瀏覽人數 <%= lcount %>

ch3/count.jsp (cont.)

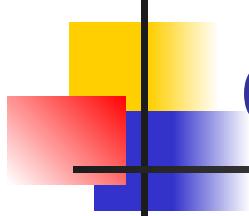
■ _count_jsp.java

變數的使用
累積瀏覽人數4, 本次瀏覽人數1

thread 只要刷新就是新的

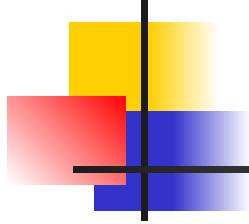
Why?

instance
只有一個



ch3/count.jsp (cont.)

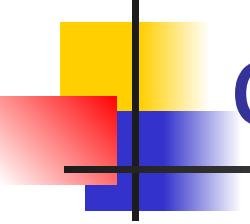
- count vs. lcount
 - 前者是 instance variable
 - 後者是 local variable
- count.jsp 只產生一個對應的 **Servlet instance**, 但以 **thread** 執行 **_jspService()** 產生每次瀏覽結果
 - count 只在 servlet instance 產生時初始化
 - 每次執行 **_jspService()** 都將 lcount 初始化



Declarations

- 當同一 JSP Page 的各連線需分享資料時，可用 **Declarations** 宣告變數

static 的概念。



ch3/stars.jsp

```
<%!
// Declarations
public void star(int i) // star() method
{
    for(;i>0;i--)
        System.out.print("*"); // 印出i個星號
}
%>
```

函數(方法)的使用
<HR>

```
<%
// Scriptlets
star(20); // 呼叫star() method
out.println("<BR><FONT SIZE = 10 COLOR = RED>Knowledge is power</FONT><BR>");
star(20); // 呼叫star() method
%>
```

在 html
↗

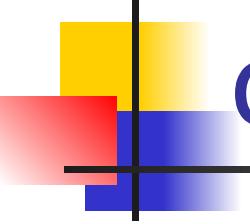
ch3/stars.jsp (cont.)

■ _stars_.jsp.java



Knowledge is power





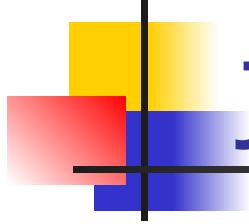
ch3/max.jsp

```
<%!
// Declarations
int score[]={9,14,6,18,2,10};
public int largest(int arr[]) {
    int max=arr[0];
    for(int i=0;i<arr.length;i++)
        if(max<arr[i])
            max=arr[i];
    return max;
}
%>
<FONT SIZE = 5 COLOR = BLUE>函數(方法)的使用</FONT>
<HR>
<%
// Scriptlets
for(int i=0; i<score.length; i++) {
%>
<%= score[i] %>
<%
}
%>
, 最大的數是<%= largest(score) %>
```

ch3/max.jsp (cont.)

■ _max_jsp.java



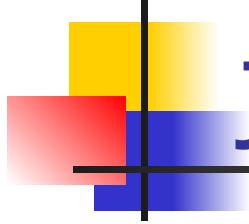


jspInit()

- 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() {...}

override



jspDestroy()

resin / tomcat

- 當 JSP Container 移除對應 Servlet instance, 釋放所用資源時執行

- 定義在 **Declarations** 中, 語法如下

public void jspDestroy() {...}

- Example

- InitDes.jsp

ch3/InitDes.jsp

```
<%@ page contentType="text/html; charset=Big5"%>
<%
public void jspInit()
{
    System.out.println("jspInit is called!");
}

public void jspDestroy()
{
    System.out.println("jspDestroy is called!");
}
%>
<HTML>
<HEAD><TITLE>jspInit函數與jspDestroy函數的使用</TITLE></HEAD>
<BODY>
<CENTER>
<FONT SIZE = 5 COLOR = blue>jspInit函數與jspDestroy函數的使用</FONT>
</CENTER>
<HR><BR>
</BODY>
</HTML>
```

ch3/InitDes.jsp (cont.)



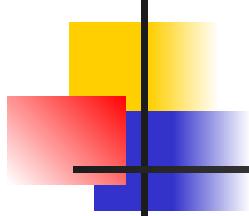
jspInit函數與jspDestroy函數的使用



ch3/InitDes.jsp (cont.)

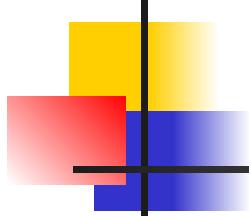
The screenshot shows a Windows desktop environment. In the top-left corner, there is a small window titled "Resin Web Server" with buttons for "Start", "Stop", and "Quit". To its right is a Firefox browser window displaying the URL "127.0.0.1:8080/ch3/ex5-8/InitDes.jsp". The page content is a single line of text: "jspInit函數與jspDestroy函數的使用". Below the browser is a terminal window with a black background and white text, showing Java system logs. A red oval highlights the line "jspInit is called!" at the bottom of the log.

```
[09:22:25.598] {main} Windows 10 10.0 x86
[09:22:25.614] {main} Java(TM) SE Runtime Environment 1.8.0_152-b16, MS950, zh
[09:22:25.614] {main} Java HotSpot(TM) Client VM 25.152-b16, 32, mixed mode, Oracle Corporation
[09:22:25.614] {main} user.name: admin
[09:22:25.614] {main} resin.home = G:\yu\course\NetPrg\2021\resin-pro-3.1.6
[09:22:25.614] {main} resin.root = G:\yu\course\NetPrg\2021\resin-pro-3.1.6
[09:22:25.614] {main} resin.conf = /G:/yu/course/NetPrg/2021/resin-pro-3.1.6/conf/resin.conf
[09:22:25.614] {main}
[09:22:25.754] {main} WebApp[http://localhost:8080] active
[09:22:25.960] {main} WebApp[http://localhost:8080/resin-admin] active
[09:22:25.969] {main} WebApp[http://localhost:8080/ch2] active
[09:22:25.979] {main} WebApp[http://localhost:8080/ch3] active
[09:22:25.987] {main} WebApp[http://localhost:8080/ch4] active
[09:22:26.280] {main} WebApp[http://localhost:8080/resin-doc] active
[09:22:26.280] {main} Host[] active
[09:22:26.283] {main} com.caucho.config.ConfigException: Resin Professional has not found any valid licenses.
[09:22:26.283] {main} Licenses belong in G:\yu\course\NetPrg\2021\resin-pro-3.1.6\licenses.
[09:22:26.283] {main} See http://www.caucho.com/sales for licensing information.
[09:22:26.324] {main} hmux listening to iccert.nhi.gov.tw:6800
[09:22:26.326] {main} http listening to *:8080
[09:22:26.326] {main} Server[id=,cluster=app-tier] active
[09:22:26.326] {main} Resin started in 1806ms
[09:22:26.460] {main} (http--8080-5) Loading .tld files from global classpath
[09:23:05.431] {Thread-4} warning: [options] bootstrap class path not set in conjunction with -source 1.5
[09:23:05.431] {Thread-4} warning: [options] source value 1.5 is obsolete and will be removed in a future release
[09:23:05.431] {Thread-4} warning: [options] To suppress warnings about obsolete options, use -Xlint:-options.
[09:23:05.431] {Thread-4} 3 warnings
jspInit is called!
```



Directives

- 設定 JSP Page 有關輸出方式、引用套件、載入檔案等相關設定
- 本身不會輸出任何網頁結果至瀏覽器，但可能會影響內容
- 有效範圍僅限於該 JSP Page
- 三種 Directives
 - page
 - include
 - taglib

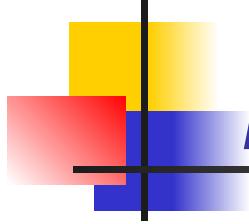


Directives (cont.)

- 語法如下

```
<%@ directive attribute1="value1"  
attribute2="value2" ... %>
```

- 設定值均為字串必須用"標示



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

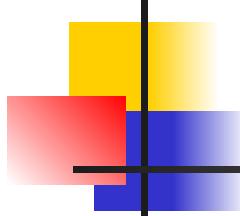
Java

$\xrightarrow{\text{之後會提到不用 JAVA}}$

- 語法如下

Action 用 XML 格式

```
<%@ page language="描述語言" %>
```



常見 attributes in *page* (cont.)

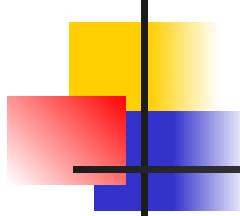
- **extends**

- 指定 JSP 產生的 Servlet 繼承的父類別
 - 語法如下

`<%@ page extends="父類別名稱" %>`

- Example

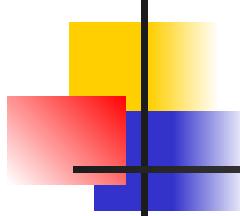
`<%@ page extends="HttpServlet" %>`



常見 attributes in *page* (cont.)

■ import

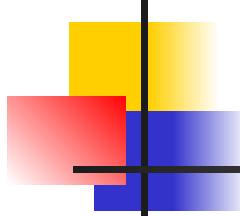
- 設定 JSP Page 使用的 Java 類別套件，可多次設定，且累加每個設定
- 語法如下
 - `<%@ page import="套件1, 套件2, ..." %>`
- Example
 - `<%@ page import="java.io.*, java.util.Date" %>`



常見 attributes in *page* (cont.)

■ session

- 設定是否使用代表使用者連線的 **session** 物件
- 確認使用者有沒有去過上一個網頁
- 語法如下
 - <%@ page session="true|false" %>
 - 預設值為 true



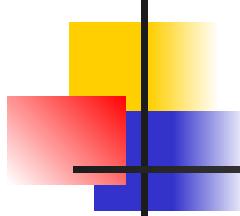
常見 attributes in *page* (cont.)

- buffer

- 網頁輸出是否使用緩衝區
- 語法如下

`<%@ page buffer="none|緩衝區大小kb" %>`

- 緩衝區大小預設值因 JSP Container 而異, Resin的預設值為8kb
 - 緩衝區大小必須是8kb的倍數
- Example
 - `<%@ page buffer="16kb" %>`



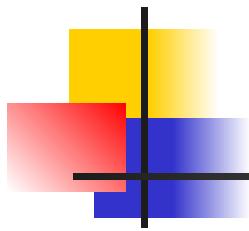
常見 attributes in *page* (cont.)

■ autoFlush

- 當緩衝區滿後，是否自動輸出緩衝區內資料
- 語法如下

`<%@ page autoFlush="true|false" %>`

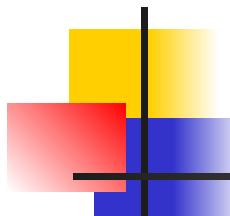
- 預設值為 true



常見 attributes in *page* (cont.)

- **isThreadSafe**

- JSP Container 對於多個客戶端請求，決定採用按請求到達順序逐一處理(**false**)，或同時處理(**true**)
- 語法如下
 - <%@ page isThreadSafe = "true|false" %>
- 預設值為 **true**
- 設定為 **false** 也不保證執行結果正確呵呵



常見 attributes in page (cont.)

- info

- 設定網頁的說明資訊
- 語法如下

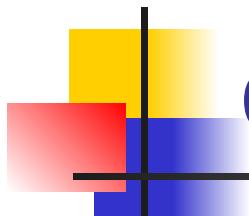
直接寫，不是另一份檔案

`<%@ page info="網頁說明資訊" %>`

- 呼叫 `getServletInfo()` 可取得此資訊
 - `<%= getServletInfo() %>`
- Example
 - [info.jsp](#)

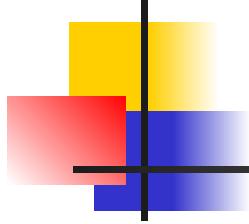
ch3/info.jsp





ch3/info.jsp (cont.)

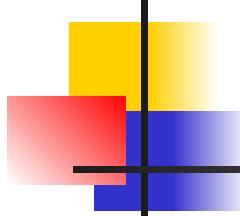
```
<%@ page contentType="text/html; charset=Big5"
    info="Demo for page attributes and setup"
%>
<HTML>
<BODY>
<!--下面的敘述將取得網頁指令所設定的info屬性-->
<%= getServletInfo() %>
</BODY>
</HTML>
```



常見 attributes in page (cont.)

■ errorPage

- 執行 JSP Page 時，若發生錯誤，設定處理錯誤訊息網頁的 URL
- 語法如下(該網頁errorPage = True)
`<%@ page errorPage="處理錯誤訊息網頁的
URL" %>犯錯設集中在 errorPage`
- 錯誤訊息以 Throwable object 表示，藉 throw 丟出 ~~errorPage 也有 isErrorPage Attribute~~
- 被設為處理該錯誤訊息的 JSP Page 則利用 exception 隱含物件取得該錯誤訊息



常見 attributes in *page* (cont.)

■ isErrorPage

- JSP Page 是否為處理錯誤訊息的網頁
- 語法如下

`<%@ page isErrorPage="true|false" %>`

- 預設值為 false
- 設定為 true 時，可存取隱含的 exception 物件，並得到發生錯誤網頁傳出的錯誤訊息
 - `<%= exception.getMessage() %>`
 - `<%= exception.toString() %>`
- Examples
 - errorPage.jsp; errorPage.jsp ; isErrorPage.jsp

ch3/errorPage.jsp



錯誤處理網頁的指定

除以



上午 09:49
2021/3/10

ch3/errorPage.jsp (cont.)

```
<%@ page contentType="text/html; charset=Big5"
   errorPage="isErrorPage.jsp" %>
<HTML>
<HEAD>
<TITLE>錯誤處理網頁的指定</TITLE>
</HEAD>
<BODY>
<CENTER>
<FONT SIZE = 5 COLOR = blue>錯誤處理網頁的指定</FONT>
</CENTER>
<HR><P></P>
<%
String strdividend = request.getParameter("dividend");
String strdivisor = request.getParameter("divisor");

if (strdividend == null || strdivisor == null )
{
    %
    <Form action="errorPage.jsp" method=post>
        <P><INPUT name=dividend>
            除以<INPUT name=divisor></P>
        <INPUT name=submit1 type=submit value=送出>
    </Form>
    %
}
.
```

ch3/errorPage.jsp (cont.)

```
else
{
    try{
        int dividend = Integer.parseInt(strdividend);
        int divisor = Integer.parseInt(strdivisor);
    %>
    <H3>答案為<Font color = red>
    <%= dividend / divisor%></Font></H3>
    <%
    }
    catch(NumberFormatException nfex)
    {
        throw new NumberFormatException("您輸入的不是整數");
    }
    catch(RuntimeException rtex)
    {
        throw new RuntimeException("不可輸入0");
    }
}
%>

</BODY>
</HTML>
```

ch3/isErrorPage.jsp

還是在errorPage 但是會執行isErrorPage



A screenshot of a Firefox browser window. The title bar says "錯誤處理網頁的建立". The address bar shows the URL "127.0.0.1:8080/ch3/errorPage.jsp". The page content is titled "錯誤處理網頁的建立" and contains the text "java.lang.RuntimeException: 不可輸入0". The browser interface includes standard buttons like back, forward, and search, along with a toolbar and a status bar at the bottom.

錯誤處理網頁的建立

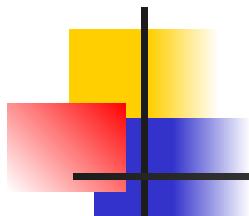
java.lang.RuntimeException: 不可輸入0



上午 09:50
2021/3/10 2

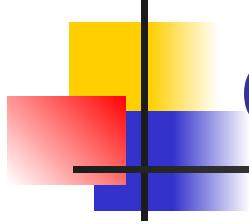
ch3/isErrorPage.jsp (cont.)

```
<%@ page contentType="text/html; charset=Big5"
    isErrorPage = "true" %>
<HTML>
<HEAD>
<TITLE>錯誤處理網頁的建立</TITLE>
</HEAD>
<BODY>
<CENTER>
<FONT SIZE = 5 COLOR = blue>錯誤處理網頁的建立</FONT>
</CENTER>
<HR><P></P>
<%= exception.toString() %>
</BODY>
</HTML>
```



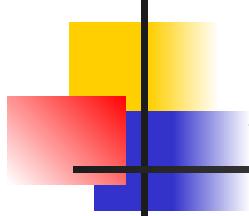
常見 attributes in *page* (cont.)

- **contentType** 可以輸出圖片、文件...
 - 設定 JSP 輸出資料使用的 MIME 型別及字元編碼方式
 - Multipurpose Internet Mail Extensions
 - 語法如下
 - <%@ page contentType="MIME型別" %>
 - <%@ page contentType="MIME型別; charset=字元編碼" %>
 - 預設值為 text/html; charset=ISO-8859-1



contentType (cont.)

- Example
 - <%@ page
contentType="text/html;charset=Big5" %>
 - 可以設成**GIF, JPEG**去顯示



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

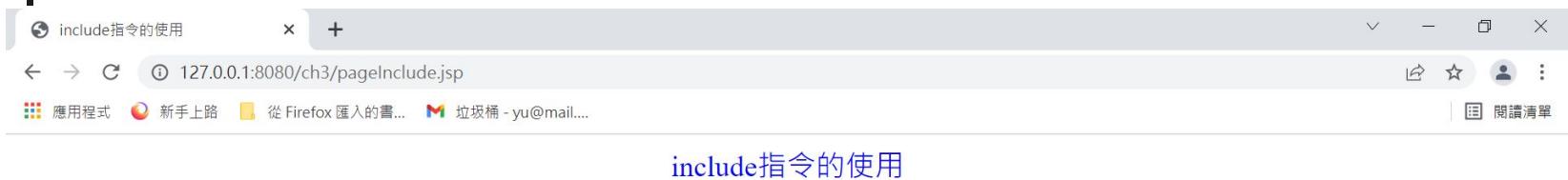
`<%@ include file="檔案名稱" %>`

- Examples

- `<%@ include file="heading.inc" %>`
 - pageInclude.jsppageInclude.jsp ;heading.in
c

不可存取→安全

ch3/pageInclude.jsp



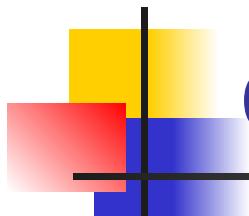
第一次載入

這是載入的檔案
這是 JSP 程式

第二次載入

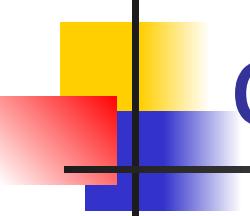
這是載入的檔案
這是 JSP 程式





ch3/pageInclude.jsp (cont.)

```
<%@ page contentType="text/html; charset=Big5" %>
<HTML>
<HEAD>
<TITLE>include指令的使用</TITLE>
</HEAD>
<BODY>
<CENTER>
<FONT SIZE = 5 COLOR = BLUE>include指令的使用</FONT>
</CENTER>
<HR>
<P>第一次載入</P>
<%@ include file = "heading.inc" %>
<p>第二次載入<p>
<%@ include file = "heading.inc" %>
</BODY>
</HTML>
```

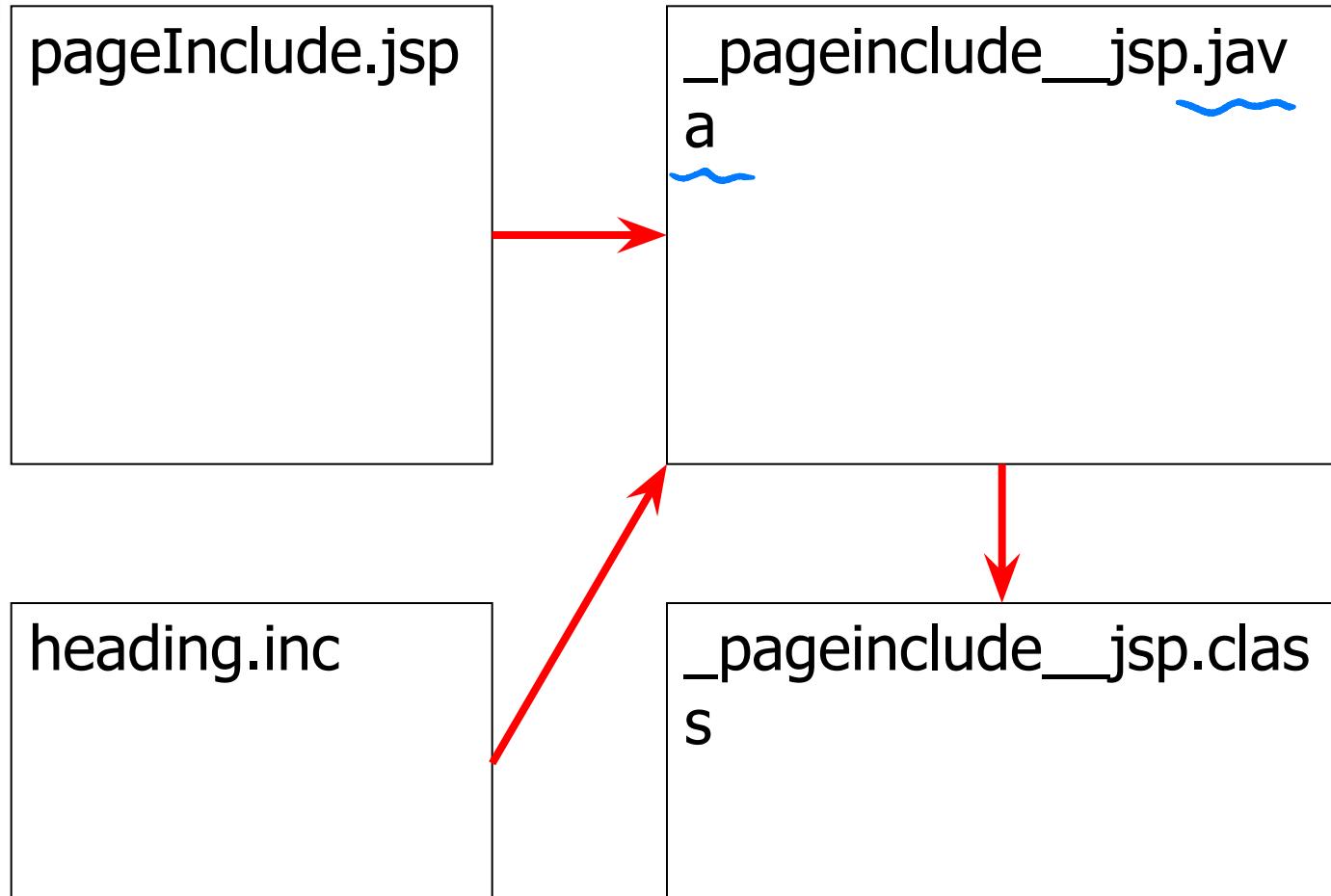


ch3/heading.inc

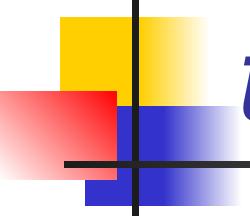
```
<FONT SIZE = 5 COLOR = Red>  
這是載入的檔案<br>  
<% out.println("這是 JSP 程式"); %>  
</FONT>
```

**.inc 不能執行，預防resin看到.jsp就拿來跑
可以放在不能瀏覽的目錄底下
加強資安問題**

ch3/pageInclude.jsp (cont.)

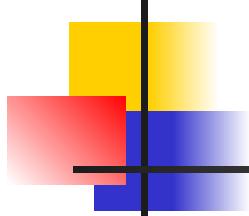


Translation Phase



taglib directive

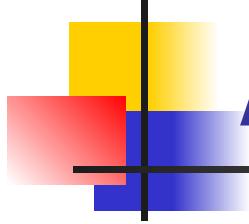
- 允許 JSP Page 使用標籤庫(tag library)
 - Collected actions that encapsulate some functionality to be used from within a JSP page
- 語法如下
 - `<%@ taglib uri="標籤庫URI" prefix="前置標記" %>`
 - Uniform Resource Identifier
 - `<%@ taglib tagdir="標籤庫目錄" prefix="前置標記" %>`



taglib directive (cont.)

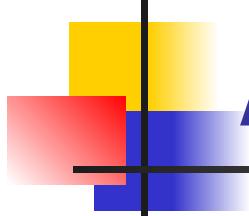
- Examples

- `<%@ taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core" %>`
- `<%@ taglib prefix="my"
tagdir="/WEB-INF/tags" %>`



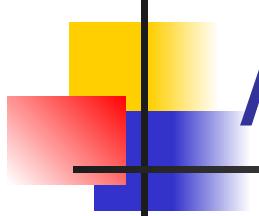
Actions

- 可建立、修改或使用 objects, 並影響輸出結果 (可以使用別人寫好的class, 使用他的instance)
 - JSP standard actions
 - Actions added by tag extensions 前面有#
- 讓 JSP 可不使用 Java 語法, 改用 XML 撰寫, 使用者可以不用會JAVA (可以用 scriplet, 但是要使用JAVA語法)



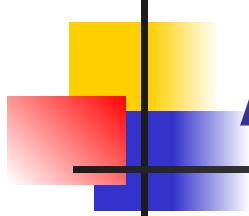
Actions (cont.)

- JSP standard actions
 - <jsp:useBean>
 - <jsp:setProperty>
 - <jsp:getProperty>
 - <jsp:param>
 - <jsp:include>
 - <jsp:forward>
 - <jsp:plugin>



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

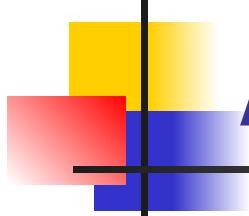


Actions (cont.)

- **<jsp:param>**

- 用於傳遞參數，搭配 **<jsp:include>**、
<jsp:forward> 或 **<jsp:plugin>** 一起使用
 - 語法如下

<jsp:param name="參數名稱" value="值" />

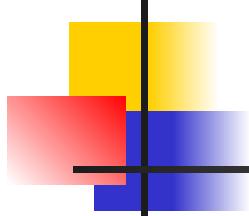


Actions (cont.)

- <jsp:include>

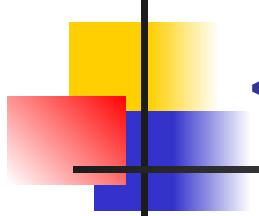
- 載入 HTML 或 JSP Page
 - 語法如下

```
<jsp:include page="網頁名稱">
<jsp:param name="參數名稱1" value="值1" />
<jsp:param name="參數名稱2" value="值2" />
...
</jsp:include>
```



<jsp:include>

- 不傳遞參數
 - <jsp:include page="網頁名稱" />**
 - 載入 HTML 網頁不需傳入參數
- 載入 JSP Page
 - 副檔名不建議用 .jsp, 可考慮 .jspf、.jsf、.inc 等自訂名稱
 - 被載入 JSP 可用以下方式取得傳入參數
 - `request.getParameter("參數名稱");`



<jsp:include> (cont.)

- Examples

- jspInclude.jsp ; JSPInc.jspf ; jspInclude.jsp ; JSPInc.jspf ; HtmlInc.htm

ch3/jspInclude.jsp



The screenshot shows a Firefox browser window with the title bar reading "<jsp:include>的使用". The address bar shows the URL "127.0.0.1:8080/ch3/jspInclude.jsp". The page content is titled "<jsp:include>的使用". It contains two sections: one for JSP and one for HTML, both demonstrating the <jsp:include> tag.

使用<jsp:include>載入的JSP網頁
Hi! My name is Shung-Chun Kuo.
I am 27 years old.
This is a JSP page.

使用<jsp:include>載入的HTML網頁
Hi! I am Shung-Chun Kuo.
I am 27 years old.
This is a HTML page.



上午 09:52
2021/3/10 2(2)

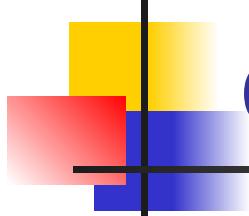
ch3/jspInclude.jsp (cont.)

```
<%@ page contentType="text/html; charset=Big5" %>
<HTML>
<HEAD>
<TITLE>< jsp:include >的使用</TITLE>
</HEAD>
<BODY>
<CENTER>
<FONT SIZE = 5 COLOR = blue>< jsp:include >的使用</FONT>
</CENTER>
<HR>

使用<Font color = red>< jsp:include ></Font>載入的JSP網頁<BR>
<jsp:include page="JSPIInc.jspf">
    <jsp:param name="name" value="Shung-Chun Kuo"/>
    <jsp:param name="age" value="27"/>
</jsp:include>
<P></P>

使用<Font color = red>< jsp:include ></Font>載入的HTML網頁<BR>
<jsp:include page="HTMLInc.htm"/>

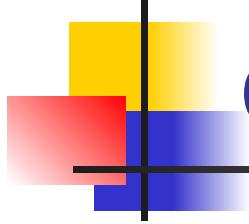
</BODY>
</HTML>
```



ch3/JSPInc.jspf

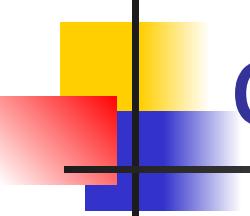
expression

```
<HTML>
<BODY>
Hi! My name is <Font Color=Blue>
<%= request.getParameter("name")%></Font>. <BR>
I am <Font Color=Blue>
<%= request.getParameter("age")%></Font> years old. <BR>
This is a <Font Color=Red>JSP</Font> page.
</BODY>
</HTML>
```



ch3/HTMLInc.htm

```
<HTML>
<BODY>
Hi! I am <Font Color=Blue>Shung-Chun Kuo</Font>.<BR>
I am <Font Color=Blue>27</Font> years old.<BR>
This is a <Font Color=Red>HTML</Font> page.
</BODY>
</HTML>
```



ch3/jspInclude.jsp (cont.)

jspInclude.jsp

_jspinclude_jsp.java

```
_jspService(...) {  
...  
  
pageContext.include("JSPInc.jspf",  
...);  
  
...}
```

_jspinclude_jsp.class

JSPInc.jspf

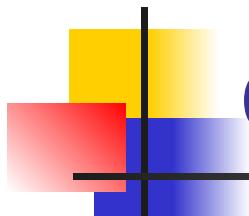
_jspinc_jspf.java

```
pageContext.include("HTMLInc.htm",  
...);  
...  
}
```

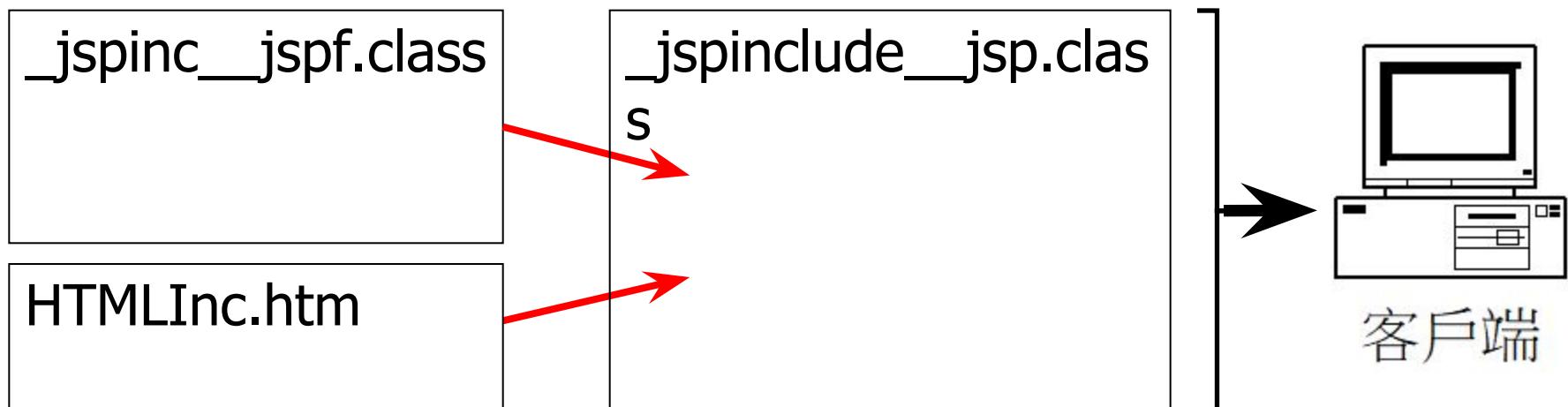
_jspinc_jspf.class

HTMLInc.htm

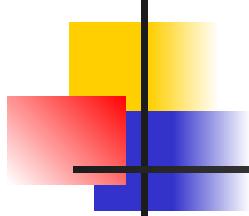
Translation Phase



ch3/jspInclude.jsp (cont.)



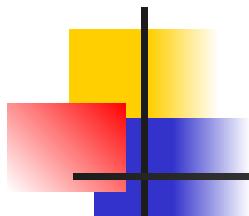
Execution Phase



<jsp:forward>

- 中止當下瀏覽網頁的執行，載入第一個 forwarded 的 HTML 或 JSP Page
- 語法如下

```
<jsp:forward page="網頁名稱">
<jsp:param name="參數名稱1" value="值1" />
<jsp:param name="參數名稱2" value="值2" />
...
</jsp:forward>
```



<jsp:forward> (cont.)

- 不傳遞參數
`<jsp:forward page="網頁名稱" />`
 - 載入 HTML 網頁不需傳入參數
- 被載入 JSP 可用以下方式取得傳入參數
 - `request.getParameter("參數名稱");`
- Examples
 - forward.jsp forward.jsp; JSPInc.jsp
- Buffer 超過會產生錯誤
forward 前面不可以太多資料

ch3/forward.jsp



A screenshot of a Firefox browser window. The address bar shows the URL `127.0.0.1:8080/ch3/forward.jsp`. The main content area displays the following text:

```
Hi! My name is Shung-Chun Kuo.  
I am 27 years old.  
This is a JSP page.
```

The browser interface includes standard navigation buttons (back, forward, search), a toolbar with various icons, and a status bar at the bottom showing system information like date and time.



上午 09:53
2021/3/10

ch3/forward.jsp (cont.)

```
<%@ page contentType="text/html; charset=Big5" %>
<HTML>
<HEAD>
<TITLE>< jsp:forward >的使用</TITLE>
</HEAD>
<BODY>
<CENTER>
<FONT SIZE = 5 COLOR = blue>< jsp:forward >的使用</FONT>
</CENTER>
<HR>
```

使用< jsp:forward >將瀏覽器導向JSP網頁

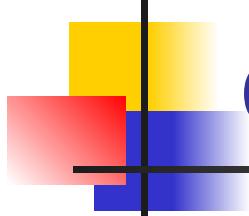

```
<jsp:forward page="JSPInc.jsp">
    <jsp:param name="name" value="Shung-Chun Kuo"/>
    <jsp:param name="age" value="27"/>
</jsp:forward>
```

<P></P>

以下就不會顯示了

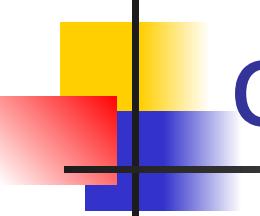
使用< jsp:forward >將瀏覽器導向HTML網頁


```
<jsp:forward page="HTMLInc.htm"/>
</BODY>
</HTML>
```



ch3/JSPInc.jsp

```
<HTML>
<BODY>
Hi! My name is <Font Color=Blue>
<%= request.getParameter("name")%></Font>. <BR>
I am <Font Color=Blue>
<%= request.getParameter("age")%></Font> years old. <BR>
This is a <Font Color=Red>JSP</Font> page.
</HTML>
</BODY>
```



ch3/forward.jsp (cont.)

forward.jsp

_forward_jsp.java

```
_jspService(...) {  
...  
  
pageContext.forward("JSPInc.jsp",  
...);  
  
...  
  
pageContext.forward("HTMLInc.htm",  
...);  
}
```

_forward_jsp.class

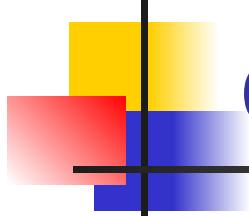
JSPInc.jsp

_jspinc_jsp.java

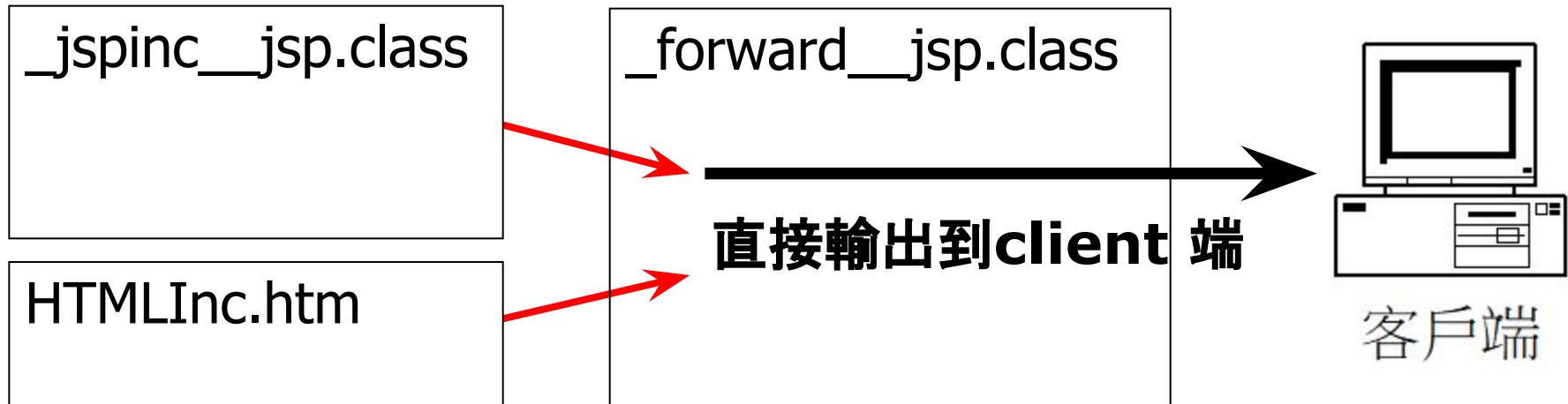
_jspinc_jsp.class

HTMLInc.htm

Translation Phase



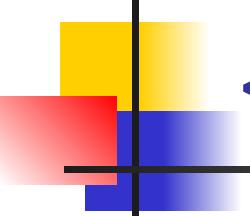
ch3/forward.jsp (cont.)



Execution Phase

include directive、<jsp:include> 及 <jsp:forward> 比較

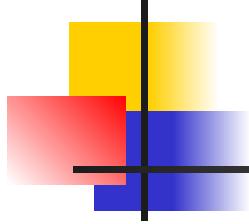
	include directive	<jsp:include>	<jsp:forward>
指令類別	Directive	Action	Action
可多個 included files?	Y	Y	Y (語法可支援, 但只有第一個有效)
Included file 是否被編譯產生 Servlet	N <i>放到 .java</i>	Y (當 included files 為 JSP)	Y (當第一個 included file 為 JSP)
Including file 執行 included file 的方式	將 included files 併入 including file 中	Including file 產生的 .java 檔中, 以 pageContext.include() 呼叫 included files, 且必須包含於 _jspService()	Including file 產生的 .java 檔中, 以 pageContext.forward() 呼叫 included files, 且必須包含於 _jspService()
執行結果	Included files 及 including file 的執行結果均會出現於瀏覽器	Included files 及 including file 的執行結果均會出現於瀏覽器, 類似呼叫函式	中止 including file 的執行, 瀏覽器僅出現第一個 included file 的執行結果, 類似 exec()
使用時特別注意事項		無法異動回應狀態碼或設定 headers, 會忽略相關 methods 的呼叫	若程式異動 ServletResponse 物件, 可能發生錯誤, 例如將 page buffer="none"



<jsp:plugin>

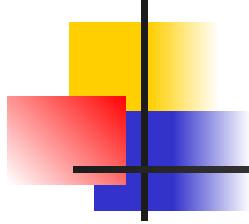
- 載入 Java Applet 或 Bean 至瀏覽器
- 與 HTML 語法的 <Applet> 及 <Object> 標籤相同
- 語法如下

```
<jsp:plugin type="plugin類型"  
code="儲存類別的檔案名稱"  
codebase="類別路徑"  
{align="對齊方式"}>
```



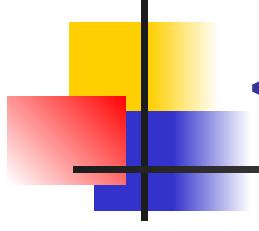
<jsp:plugin> (cont.)

```
{archive="相關檔案路徑"}  
{height="高度"}  
{width="寬度"}  
{hspace="水平間距"}  
{vspace="垂直間距"}  
{jrevesion="Java環境版本"}  
{name="物件名稱"}  
{nspluginurl="供NC使用的plugin載入位置"}  
{iepluginurl="供ie使用的plugin載入位置"}>
```



<jsp:plugin> (cont.)

```
{<jsp:params>
  <jsp:param name="參數名稱1" value="值1" />
  <jsp:param name="參數名稱2" value="值2" />
  ...
</jsp:params>}
{<jsp:fallback>錯誤訊息</jsp:fallback>}
</jsp:plugin>
```



<jsp:plugin> (cont.)

- Examples
 - plugin.jsp; plugin.jsp; HelloWorld.java

ch3/plugin.jsp

The screenshot shows a Windows Internet Explorer window displaying a JSP page titled "<jsp:plugin>的使用". The page content includes two sections: one using the <applet> tag and another using the <jsp:plugin> tag. Both sections output the text "Hello World! I am Shung-Chun Kuo!".

使用`<applet>`標籤載入的Applet

Hello World! I am Shung-Chun Kuo!

使用`<jsp:plugin>`載入的Applet

Hello World! I am Shung-Chun Kuo!

完成

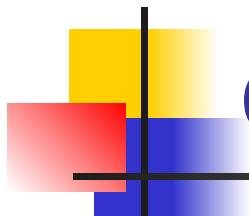
105%

ch3/plugin.jsp (cont.)

```
<%@ page contentType="text/html; charset=Big5"%>
<HTML>
<HEAD><TITLE>< jsp:plugin >的使用</TITLE></HEAD>
<BODY>
<CENTER>
<FONT SIZE = 5 COLOR = blue>< jsp:plugin >的使用</FONT>
</CENTER>
<HR>
<BR>
使用<Font color = red>Applet</Font>標籤載入的Applet<BR>
<applet code="HelloWorld.class" height=40 width=320>
    <param name=name value="Shung-Chun Kuo">
</applet>
<P></P>

使用<Font color = red>< jsp:plugin ></Font>載入的Applet<BR>
<jsp:plugin type="applet" code="HelloWorld.class" height="40" width="320">
    <jsp:params>
        <jsp:param name="name" value="Shung-Chun Kuo"/>
    </jsp:params>
    <jsp:fallback>無法載入Applet</jsp:fallback>
</jsp:plugin>

</BODY>
</HTML>
```



ch3/HelloWorld.java

```
import java.applet.Applet;
import java.awt.Graphics;

public class HelloWorld extends Applet {
    String name;
    public void init()
    {
        name = getParameter("name");
    }
    public void paint(Graphics g) {
        g.drawString("Hello World! I am " + name + "!", 50, 25);
    }
}
```