

Java魔法營

高級(畫龍點睛)

揭聲 capricorn.rich.cat@gmail.com

Java Web

應用與實務

- JSP(Java Server Page)
 - JSP model
 - EL
 - JSP standard action
 - JSP tag library
 - Custom Tags

Java Web

應用與實務

- JSP model
 - Lifecycle
 - JSP隱含物件
 - JSP網頁剖析

Java Web

JSP model

- Lifecycle
 - JSP page translation
 - JSP page compilation
 - load class
 - create instance
 - call void `jspInit()`
 - call void `_jspService(HttpServletRequest request, HttpServletResponse response)`
 - call void `jspDestroy()`

JSP model

Lifecycle

- JSP page translation
 - 將JSP的網頁轉譯成Servlet
 - 收到第1個request進行轉譯
 - 內容有異動會重新轉譯
- 轉譯後的Servlet
 - 放置於work目錄下
 - 繼承HttpJspBase

Java Web

JSP model

- JSP隱含物件

隱含物件名稱	型態	介面/類別
request	jakarta.servlet.http.HttpServletRequest	interface
response	jakarta.servlet.http.HttpServletResponse	interface
application	jakarta.servlet.ServletContext	interface
config	jakarta.servlet.ServletConfig	interface
session	jakarta.servlet.http.HttpSession	interface
out	jakarta.servlet.jsp.JspWriter	abstract class
pageContext	jakarta.servlet.jsp.PageContext	abstract class
page	java.lang.Object	class
exception	java.lang.Throwable	class

JSP model

JSP隱含物件

- pageContext
 - 提供存取4種scope的attribute
 - PageContext.PAGE_SCOPE
 - 僅在單一JSP網頁中
 - PageContext.REQUEST_SCOPE
 - PageContext.SESSION_SCOPE
 - PageContext.APPLICATION_SCOPE

JSP model

JSP隱含物件

- pageContext

- Object getAttribute(String name)
- void setAttribute(String name, Object value)

僅在page

- void removeAttribute(String name)
- Object findAttribute (String name)

依序page, request, session, application查找

- Object getAttribute(String name, int scope)
- void setAttribute(String name, Object value, int scope)
- void removeAttribute(String name, int scope)

指定scope

JSP model

JSP隱含物件

- pageContext

設值

取值

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>PageContext Example</title>
</head>
<body>

<%
    pageContext.setAttribute("pageAttribute", "This is a page scoped attribute.");
    pageContext.setAttribute("requestAttribute",
        "This is a request scoped attribute.", PageContext.REQUEST_SCOPE);
    pageContext.setAttribute("sessionAttribute",
        "This is a session scoped attribute.", PageContext.SESSION_SCOPE);
    pageContext.setAttribute("applicationAttribute",
        "This is an application scoped attribute.", PageContext.APPLICATION_SCOPE);
%>

<h2>Attributes set using pageContext:</h2>

<p><strong>Page Scope:</strong>
    <%= pageContext.getAttribute("pageAttribute") %></p>
<p><strong>Request Scope:</strong>
    <%= pageContext.getAttribute("requestAttribute", PageContext.REQUEST_SCOPE) %></p>
<p><strong>Session Scope:</strong>
    <%= pageContext.getAttribute("sessionAttribute", PageContext.SESSION_SCOPE) %></p>
<p><strong>Application Scope:</strong>
    <%= pageContext.getAttribute("applicationAttribute", PageContext.APPLICATION_SCOPE) %></p>

</body>
</html>
```

JSP model

JSP隱含物件

- page
 - 實際上很少用到
 - 代表當前的JSP，JSP會轉譯成Servlet，所以也是代表當前的Servlet
 - 型態為Object，所以能做的事情有限

JSP model

JSP隱含物件

- exception
 - 僅在error page中使用
 - 用來取得發生例外的JSP中，丟出的Exception
 - 在web.xml中設定Exception類型對應到的錯誤畫面

JSP model

JSP隱含物件

- exception

設定為error page

取得exception內容

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Error Example</title>
</head>
<body>
    <%
        int result = 5 / 0;
    %>
</body>
</html>
```

丟出Exception

Exception及畫面的對應

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" isErrorPage="true" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Error Handler</title>
</head>
<body>
    <h2>An error occurred:</h2>
    <p><strong>Message:</strong> <%= exception.getMessage() %></p>
    <h3>Stack Trace:</h3>
    <pre>
        <% exception.printStackTrace(new java.io.PrintWriter(out)); %>
    </pre>
</body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
        https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd"
    version="6.0"
    metadata-complete="false">
    <error-page>
        <exception-type>java.lang.Throwable</exception-type>
        <location>/error.jsp</location>
    </error-page>
</web-app>
```

JSP model

JSP隱含物件

- web.xml中的jsp設定

```
<jsp-config>
  <taglib>
    <taglib-uri>http://domain/function</taglib-uri> // tld檔中的uri位置
    <taglib-clocation>/WEB-INF/tags/my.tld<taglib-location> // tld檔案位址
  </taglib>
  <jsp-property-group>
    <url-pattern>*.jsp</url-pattern>
    <page-encoding>UTF-8</page-encoding> // 統一編碼
    <default-content-type>text/html</default-content-type> // 內容類型
    <buffer>64kb</buffer> // 緩衝區大小
    <include-prelude>header.jspf</include-prelude> // 開頭jsp
    <include-coda>footer.jspf</include-coda> // 結尾jsp
    <trim-directive-whitespaces>true</trim-directive-whitespaces> // 轉譯後不留空行
    <scripting-invalid>true</scripting-invalid> // 禁用scriptlet
    <el-ignored>true</el-ignored> // 不使用EL
    <is-xml>false</is-xml> // 使用xml剖析jsp元素
  </jsp-property-group>
</jsp-config>
```


JSP model

JSP隱含物件

- web.xml中的jsp設定

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
    https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd"
  version="6.0"
  metadata-complete="false">

  <jsp-config>
    <jsp-property-group>
      <url-pattern>*.jsp</url-pattern>
      <trim-directive-whitespaces>true</trim-directive-whitespaces>
      <include-pragma>header.jspf</include-pragma>
      <include-coda>footer.jspf</include-coda>
    </jsp-property-group>
  </jsp-config>

</web-app>
```

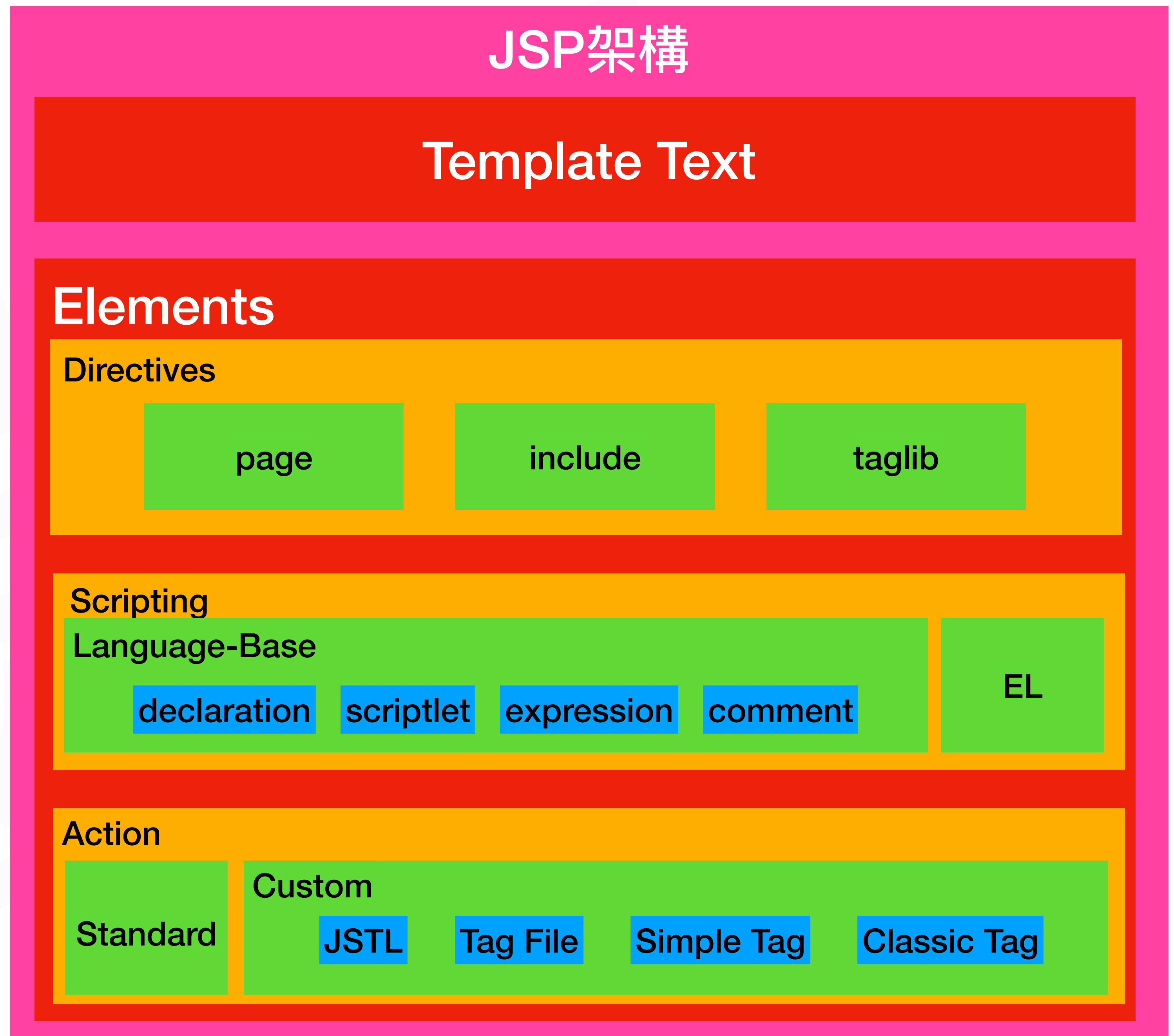
```
<!DOCTYPE html>
<html>
<head>
  <title>My Website</title>
</head>
<body>
<header>
  <h1>Welcome to My Website</h1>
</header>
```

```
<footer>
  <p>Copyright &copy; 2023 My Website</p>
</footer>
</body>
</html>
```

Java Web

JSP model

- JSP網頁剖析



JSP model

JSP網頁剖析

- Template Text
 - 轉譯階段不會有任何改變的內容
 - 通常是純HTML內容

JSP架構

Template Text

Elements

Directives

page

include

taglib

Scripting

Language-Base

declaration

scriptlet

expression

comment

EL

Action

Standard

Custom

JSTL

Tag File

Simple Tag

Classic Tag

JSP model

JSP網頁剖析

- Directives
 - 告訴JSP容器有關轉譯的指示
 - `<%@ ... %>`
 - page：頁面相關的指示
 - include：包含另一個JSP
 - taglib：使用自定標籤

JSP架構

Template Text

Elements

Directives

page

include

taglib

Scripting

Language-Base

declaration

scriptlet

expression

comment

EL

Action

Standard

Custom

JSTL

Tag File

Simple Tag

Classic Tag

JSP model

JSP網頁剖析

- Directives
 - page
 - import : 導入Java的class或package
 - language : 設定JSP的腳本語言，預設為java
 - session : 是否使用session，預設為true
 - contentType : 回應內容類型
 - isELIgnored : 是否忽略EL，預設為false
 - errorPage : 發生例外時要轉到哪個錯誤頁面
 - isErrorPage : 是否為錯誤頁面，預設為false
 - trimDirectiveWhitespaces : 轉譯後不留空行

JSP model

JSP網頁剖析

- Directives
 - include
 - file：被包含網頁的路徑
 - taglib
 - prefix：自定標籤的前綴字，相當於namespace
 - uri：自定標籤的URI，對應到web.xml中，不可重複

JSP model

JSP網頁剖析

- Directives

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>Main JSP Page</title>
</head>
<body>
    <h1>Welcome to the Main JSP Page</h1>
    <%@ include file="include.jsp" %>
    <p>This is the main content of the main.jsp page.</p>
</body>
</html>
```

更面相關的指示

包含另一個JSP

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<div style="background-color: #f5f5f5; padding: 10px; border: 1px solid #ddd;">
    <h2>This is the Header from header.jsp</h2>
    <p>This header can be included in multiple JSP pages.</p>
</div>
```

JSP model

JSP網頁剖析

- Language-Base
 - declaration：轉譯後放置於class中
 - `<%! ... %>`
 - scriptlet：轉譯後放置於_jspService()中
 - `<% ... %>`
 - expression：轉譯後放置於out.print()中
 - `<%= ... %>`
 - comment：jsp註解
 - `<%-- ... --%>`

JSP架構

Template Text

Elements

Directives

page

include

taglib

Scripting

Language-Base

declaration

scriptlet

expression

comment

EL

Action

Standard

Custom

JSTL

Tag File

Simple Tag

Classic Tag

JSP model

JSP網頁剖析

- Language-Base
 - 在JSP中儘可能不要有scriptlet及expression
 - declaration本身不必要
 - comment轉譯就會被移除
- 取代方式
 - EL取代expression
 - action取代scriptlet

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>JSP Example</title>
</head>
<body>
    <!-- This is a JSP comment. It will not be visible in the rendered HTML. --%>
    <!-- This is a HTML comment -->
    <%!
        int counter = 0;
        String incrementCounter() {
            counter++;
            return "Counter incremented!";
        }
    %>
    <%
        String message = "Hello from JSP!";
        String loopMessage = "";
        for(int i = 1; i <= 5; i++) {
            loopMessage += "Loop iteration " + i + "<br/>";
        }
    %>
    <p><strong>Message:</strong> <%= message %></p>
    <p><strong>Action:</strong> <%= incrementCounter() %></p>
    <p><strong>Current Counter Value:</strong> <%= counter %></p>
    <p><strong>Loop Output:</strong><br/> <%= loopMessage %></p>
</body>
</html>
```

comment

HTML註解

declaration

scriptlet

expression

Java Web

應用與實務

- EL(Expression Language)
- 基本語法
- 隱含物件
- 自定義

JSP架構

Template Text

Elements

Directives

page

include

taglib

Scripting

Language-Base

declaration

scriptlet

expression

comment

EL

Action

Standard

Custom

JSTL

Tag File

Simple Tag

Classic Tag

Java Web

Expression Language

- 基本語法
 - 使用EL
 - 運算子
 - lambda

Expression Language

基本語法

- 使用EL
 - `${expression}`
 - `expression`放置EL運算子

Expression Language

基本語法

- 運算子
 - 指派運算子
 - 算數運算子
 - 關係運算子
 - 邏輯運算子
- empty
- 取得資料

Expression Language

基本語法

- 指派運算子
 - 指派變數值
 - 多個變數可同時指派
 - 使用 ; 隔開
 - 最後一個不能加 ;
 - 會打印最後一個值
- += 是字串相接
 - 不會重新賦值

```
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
a => ${a = 1}<br/>
b => ${b = 2}<br/>
c = 3; d = 4 => ${c = 3; d = 4}<br/>
a += b => ${a += b}<br/>
a + b => ${a + b}<br/>
a => ${a}<br/>
</body>
</html>
```

指派變數值

指派多個變數

字串相接

Expression Language

基本語法

- 算數運算子

算術運算子	運算子	別名
加	+	
減	-	
乘	*	
除	/	div
餘	%	mod

```
<%@page import="java.util.*" %>
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  i = ${i = 100}<br/>
  j = ${j = 10}<br/>
  i + j = ${i + j}<br/>
  i - j = ${i - j}<br/>
  i * j = ${i * j}<br/>
  i / j = ${i div j}<br/>
  i % j = ${i mod j}<br/>
</body>
</html>
```

算數運算

Expression Language

基本語法

- 關係運算子

關係運算子	運算子	別名
大於	>	gt
小於	<	lt
等於	==	eq
大於等於	>=	ge
小於等於	<=	le
不等於	!=	ne

```
<%@page import="java.util.*" %>
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
i = ${i = 100}<br/>
j = ${j = 10}<br/>
i > j = ${i gt j}<br/>
i < j = ${i lt j}<br/>
i == j = ${i eq j}<br/>
i >= j = ${i ge j}<br/>
i <= j = ${i le j}<br/>
i != j = ${i ne j}<br/>
</body>
</html>
```

關係運算

Expression Language

基本語法

- 邏輯運算子

邏輯運算子	運算子	別名
logical "and"	&&	and
logical "or"		or
logical "not"	!	not

```
<%@page import="java.util.*" %>
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
true && false = ${true and false}<br/>
true || false = ${true or false}<br/>
!false = ${not false}<br/>
</body>
</html>
```

邏輯運算

Expression Language

基本語法

- empty
 - `${empty obj}`
 - 判斷為空
 - `obj`為空字串、空陣列或空集合時為true

```
<%@page import="java.lang.*" %>
<%@page import="java.util.*" %>
<%@page language="java" %>
<%@page session="true"%>
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <%
    pageContext.setAttribute("a", 1);
  %>
  a is empty? ${empty a}<br/>
</body>
</html>
```

empty



Expression Language

基本語法

- 取得資料

- 物件

- `${obj.name}`, `${obj["name"]}`

- 陣列與集合


- `${arr[0]}`

- Map

- `${map.a}`, `${map["a"]}`

- JSTL迴圈中可使用`${map.key}`取得key，`${map.value}`取得value

```
<%@page import="java.util.*" %>
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
<%
  List<String> list = List.of("A", "B");
  pageContext.setAttribute("list", list);
  Map<String,Integer> map = Map.of("a", 1, "b", 2);
  pageContext.setAttribute("map", map);
%>
list[0] = ${list[0]}<br/>
map.a + map.b = ${map["a"]} + map.b<br/>
</body>
</html>
```



取得list及map的值

Expression Language

基本語法

- lambda
 - 使用function interface
 - stream style

```
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  ${add = (x,y)->x+y}<br/>
  ${add(1,2)}<br/>
  ${()->add(1,2)+add(3,4)}<br/>
  ${bar = ["A", "B", "C"]}<br/>
  ${bar.stream()
    .map(b->b.toLowerCase())
    .toList()}<br/>
</body>
</html>
```

function interface

stream style

Java Web

Expression Language

- 隱含物件

物件名稱	描述	JSP對應物件
pageContext	代表JSP的PageContext物件	pageContext
pageScope	page scope屬性的Map	pageContext.getAttribute()
requestScope	request scope屬性的Map	request.getAttribute()
sessionScope	session scope屬性的Map	session.getAttribute()
applicationScope	application scope屬性的Map	application.getAttribute()
param	ServletRequest參數名稱和第一個值的Map	request.getParameter()
paramValues	ServletRequest參數名稱和所有值的Map	request.getParameterValues()
header	HttpServletRequest標頭名稱和第一個有值的Map	request.getHeader()
headerValues	HttpServletRequest標頭名稱和所有值的Map	request.getHeaders()
cookie	HttpServletRequest的cookie名稱和cookie物件的Map	request.getCookies()
initParam	ServletContext參數名稱和值的Map	config.getServletContext().getInitParameter()

Expression Language

隱含物件

- 隱含物件

取得request header

取得request parameter

取得路徑

取得session attribute

取得application attribute

```
<!DOCTYPE html>
<html>
<head>
  <title>EL Implicit Objects Example</title>
</head>
<body>
<%
  session.setAttribute("loggedIn", true);
  application.setAttribute("siteName", "MyWebsite");
%>
<h2>EL Implicit Objects</h2>

<p><strong>Request Parameter (username):</strong> ${param.username}</p>

<p><strong>User Agent:</strong> ${header['User-Agent']}</p>

<p><strong>Real Path of this JSP:</strong> ${pageContext.request.contextPath}
${pageContext.request.servletPath}</p>

<p><strong>Is User Logged In:</strong> ${sessionScope.loggedIn}</p>

<p><strong>Site Name:</strong> ${applicationScope.siteName}</p>

</body>
</html>
```

Java Web

Expression Language

- 自定義
 - 自定函式
 - 方法呼叫

Expression Language

自定義

- 自定函式
 - 撰寫類別
 - 撰寫標籤描述檔
 - JSP Directives
 - EL 語法
 - 佈署及打包

Expression Language

自定函式

- 撰寫類別

- public class
- public static method
- 要有回傳值

```
package edu.javaWeb;  
public class DemoEL{  
    public static String bar(String name){  
        return "Hi "+name;  
    }  
}
```

public class

公開類別等級方法
且有回傳值

Expression Language

自定函式

- 撰寫標籤描述檔(tld)
- tld檔放置於WEB-INF中即可
 - 通常在WEB-INF/tlds
- 若打包則放置於META-INF/tlds

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd">
  <tlib-version>1.0</tlib-version>
  <short-name>demo</short-name>
  <uri>http://demo.javaweb.edu/bar</uri>
  <function>
    <description>Demo</description>
    <name>bar</name>
    <function-class>edu.javaweb.DemoEL</function-class>
    <function-signature>String bar(java.lang.String)</function-signature>
  </function>
</taglib>
```

命名空間的識別名稱及uri

EL的呼叫名稱

要呼叫的類別
完整名稱及方法簽名

Expression Language

自定函式

- JSP Directives及EL語法
 - uri對應到tld中
 - prefix為命名空間
 - 最好對應tld中的short-name
- 透過命名空間呼叫方法
 - `${prefix:method(args)}`

```
<%@page contentType="text/html" %>
<%@taglib prefix="demo" uri="http://demo.javaweb.edu/bar" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  ${demo:bar("jason")}<br/>
</body>
</html>
```

prefix為命名空間

uri對應到tld

透過prefix呼叫方法

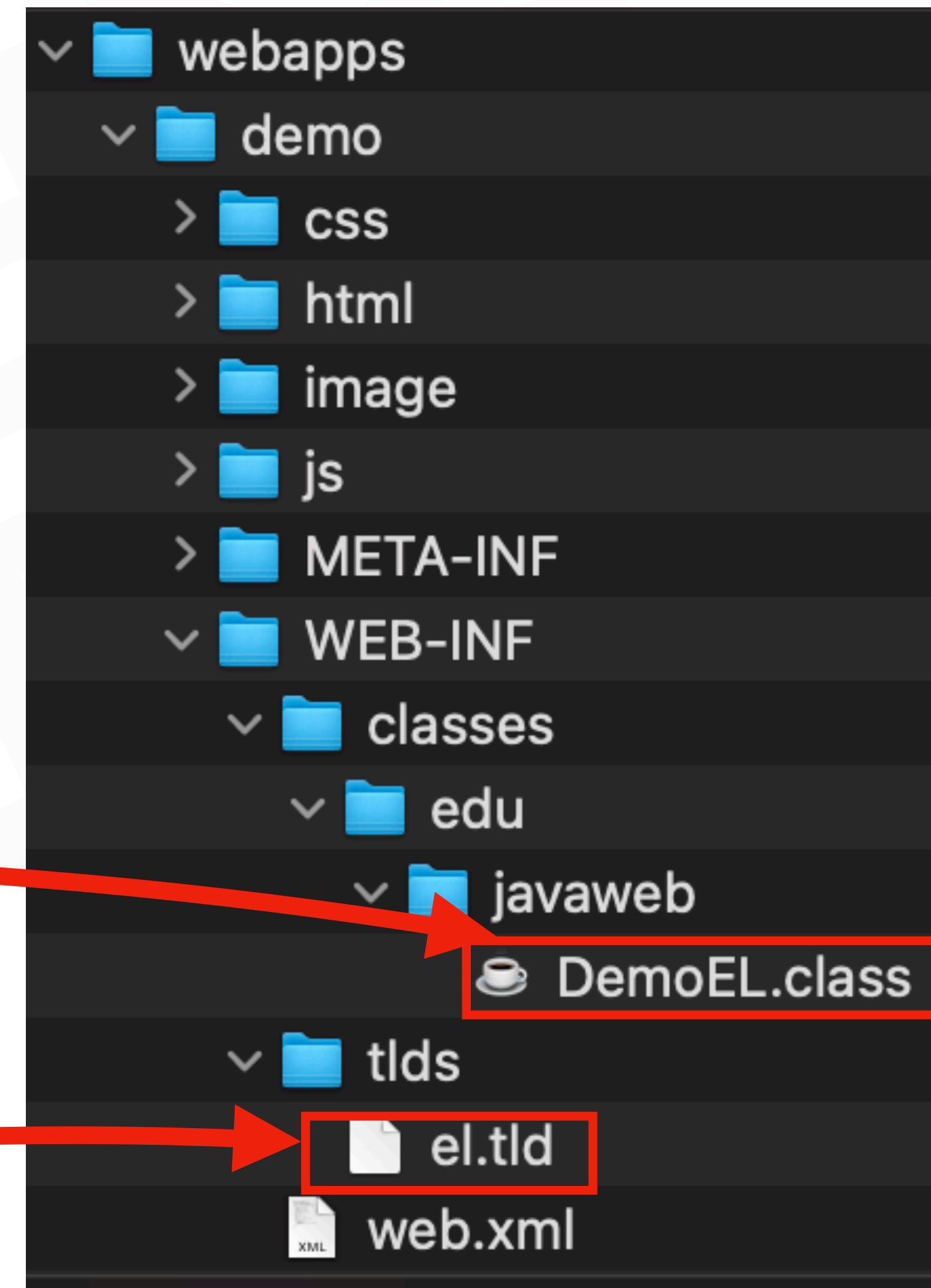
Expression Language

自定函式

- 佈署
 - 標籤描述檔(tld)放在WEB-INF下任意位置
 - 習慣上會放在WEB-INF/tlds

自定函式類別

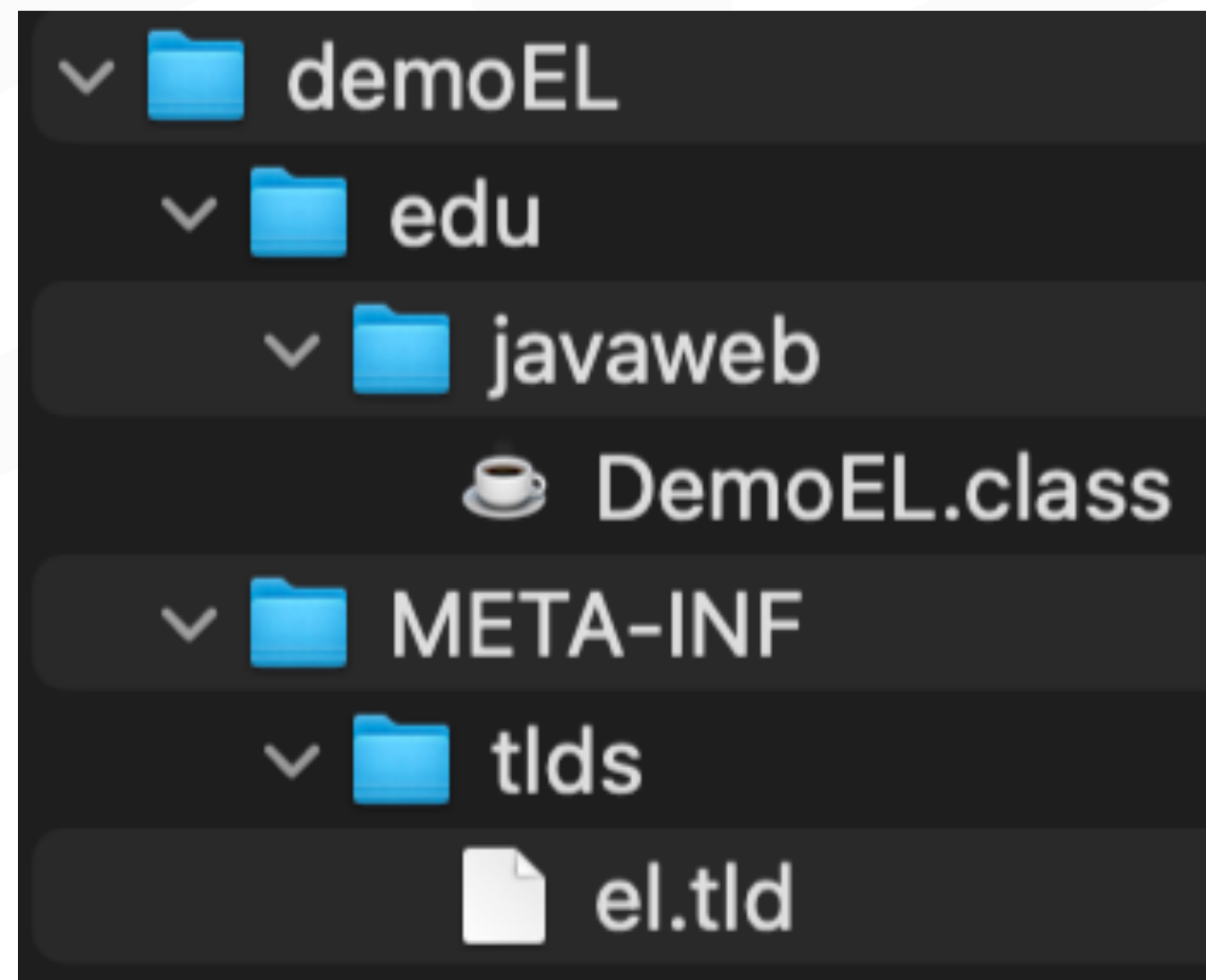
標籤描述檔



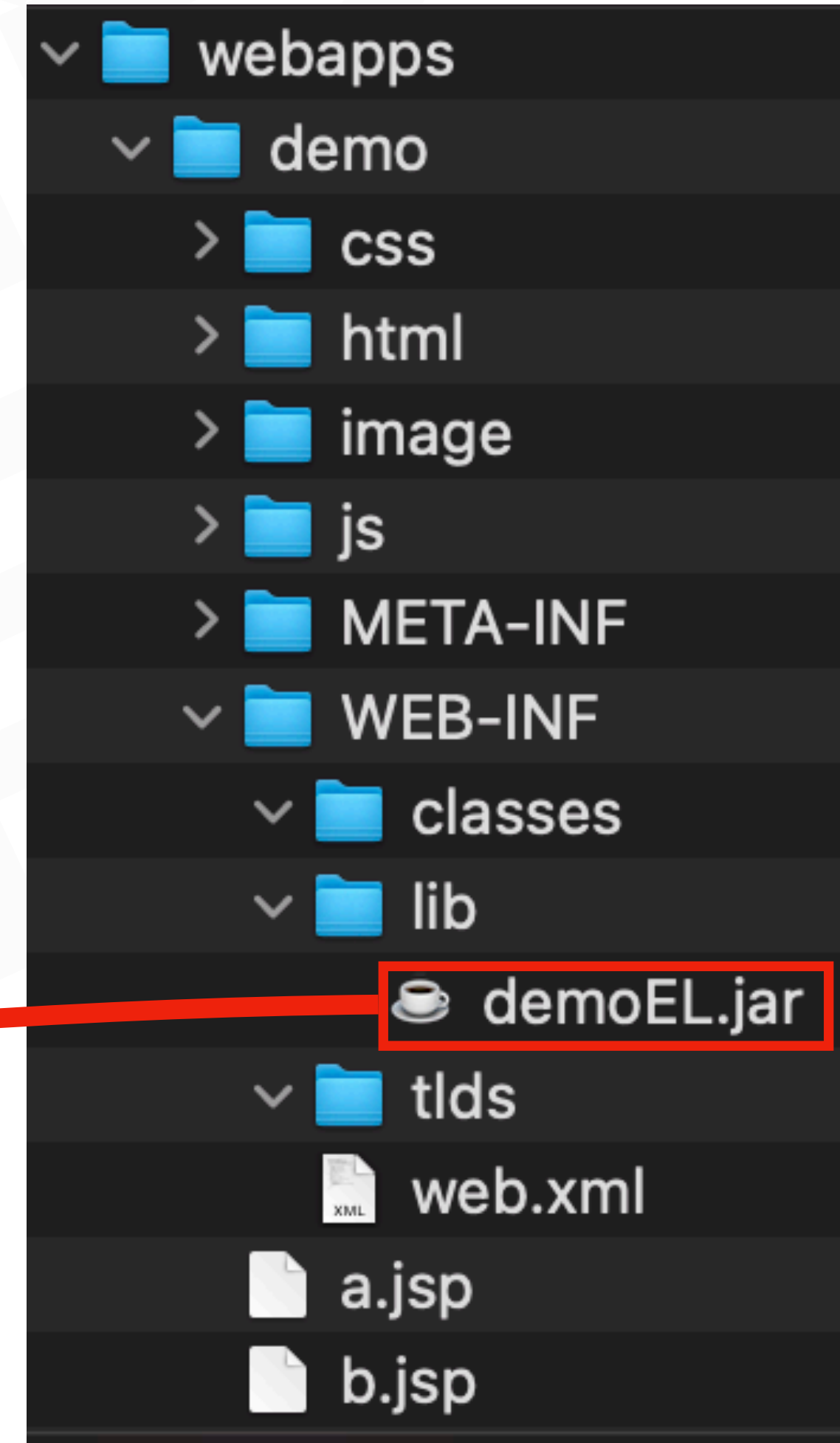
Expression Language

自定函式

- 打包
 - 標籤描述檔放置於META-INF/tlds



打包內容



Expression Language

自定義

- 方法呼叫

- 呼叫物件等級方法

- 呼叫類別等級方法

- 導入class, package, static

- void importClass(String name)

- void importPackage(String packageName)

- void importStatic(String name)

- 物件實例化不需加上new

```
<%@page contentType="text/html" %>
${pageContext.ELContext.importHandler.importClass("java.util.Date")}
<html>
<head>
  <title>Demo</title>
</head>
<body>
  name = ${name = "jason".toUpperCase()}</>
  age = ${Integer.parseInt("18")}<br/>
  now = ${Date()}
</body>
</html>
```

import

物件等級方法

類別等級方法

實例化

Java Web

應用與實務

- JSP standard action
 - POJO跟Bean
 - bean跟property
 - forward跟include

JSP架構

Template Text

Elements

Directives

page

include

taglib

Scripting

Language-Base

declaration

scriptlet

expression

comment

EL

Action

Standard

Custom

JSTL

Tag File

Simple Tag

Classic Tag

Java Web

JSP standard action

- POJO(Plain Old Java Object)
 - 普通的Java物件
 - 不受框架或規範的限制
 - 不依賴特定的技術或環境
- Bean
 - 遵循特定的命名慣例
 - 存取屬性使用setter跟getter
 - 具有無參數建構子
 - 可序列化

Bean是POJO，POJO不一定是Bean

Java Web

JSP standard action

- bean跟property
 - `<jsp:useBean>` : new一個bean物件
 - id, class, scope
 - scope的值為page(預設), request, session, application
 - `<jsp:setProperty>` : 設定bean物件的屬性值
 - name, property, value
 - `<jsp:getProperty>` : 取得bean物件的屬性值
 - name, property

JSP standard action

bean跟property

實例化

`var user = new foo.DemoUser()`

設值

`user.setName("jason")`

```
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <jsp:useBean id="user" class="edu.javaweb.DemoUser"/>
  <jsp:setProperty name="user" property="name" value="jason"/>
  Hi <jsp:getProperty name="user" property="name"/><br/>
  Hello ${user.name}
</body>
</html>
```

取值

`user.getName()`

EL取值

```
package edu.javaweb;
public class DemoUser{
  private String name;

  private int age;

  public void setName(String name){
    this.name = name;
  }

  public String getName(){
    return this.name;
  }

  public void setAge(int age){
    this.age = age;
  }

  public int getAge(){
    return this.age;
  }
}
```


JSP standard action

bean跟property

prefix

標籤名稱

屬性

屬性值

```
<jsp:forward page="include.jsp">  
  <jsp:param name="a" value="123"/>  
</jsp:forward>
```

結尾標籤

標籤本體

無標籤本體

```
<jsp:forward page="include.jsp"/>
```

Java Web

JSP standard action

- forward
 - 將request轉發至另一個JSP
 - 與RequestDispatcher.forward()相同
 - 可使用param進行param的設定
- include
 - 將另一個JSP包含進來
 - 與RequestDispatcher.include()相同
 - 可使用param進行param的設定

JSP standard action

forward

```
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
<jsp:useBean id="user" class="edu.javaweb.DemoUser"/>
<jsp:setProperty name="user" property="name" value="jason"/>
Hi <jsp:getProperty name="user" property="name"/><br/>
Hello ${user.name}
<jsp:forward page="include.jsp">
  <jsp:param name="a" value="123"/>
</jsp:forward>
</body>
</html>
```

Hello, this is include page.

\${param.a}

用EL取得參數

帶參數將需求轉發到另一個JSP

JSP standard action

include

```
<%@page contentType="text/html" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <jsp:useBean id="user" class="edu.javaweb.DemoUser"/>
  <jsp:setProperty name="user" property="name" value="jason"/>
  Hi <jsp:getProperty name="user" property="name"/><br/>
  Hello ${user.name}<br/>
  <jsp:include page="include.jsp">
    <jsp:param name="a" value="123"/>
  </jsp:include>
</body>
</html>
```

Hello, this is include page.

\${param.a}

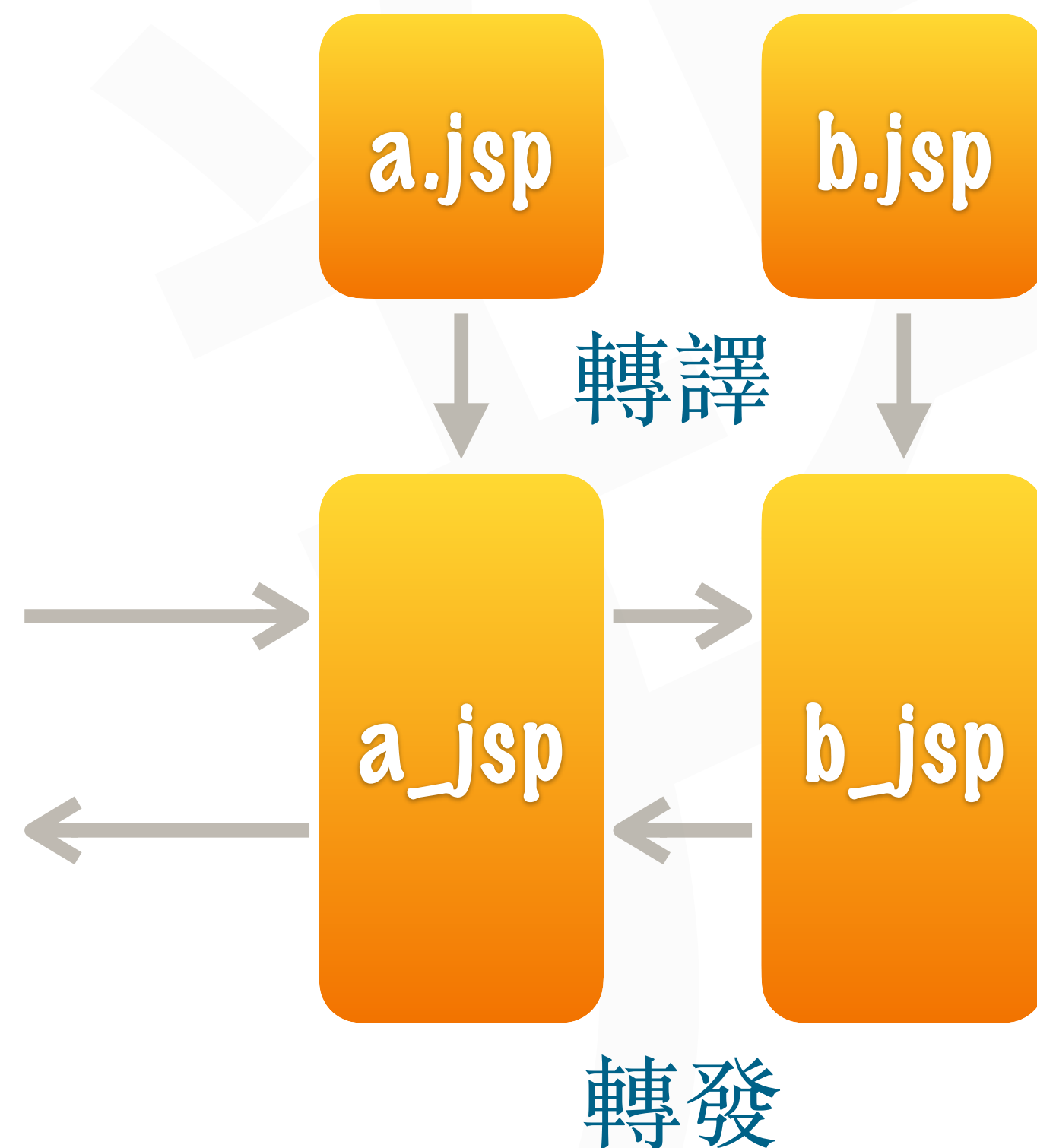
用EL取得參數

帶參數包含另一個JSP

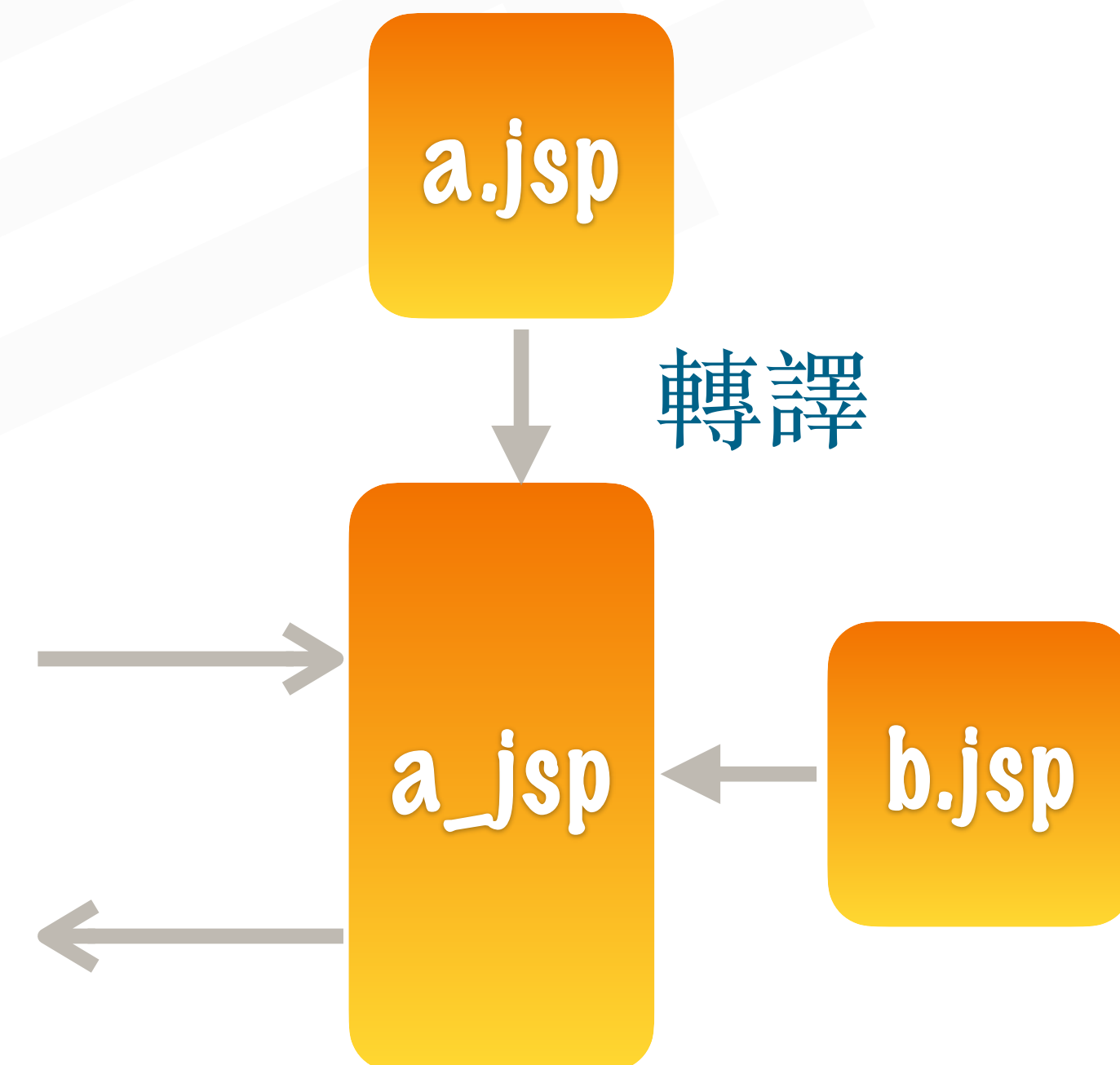
JSP standard action

include

- standard action include



- directives include



JSP standard action

include

- standard action include
- directives include

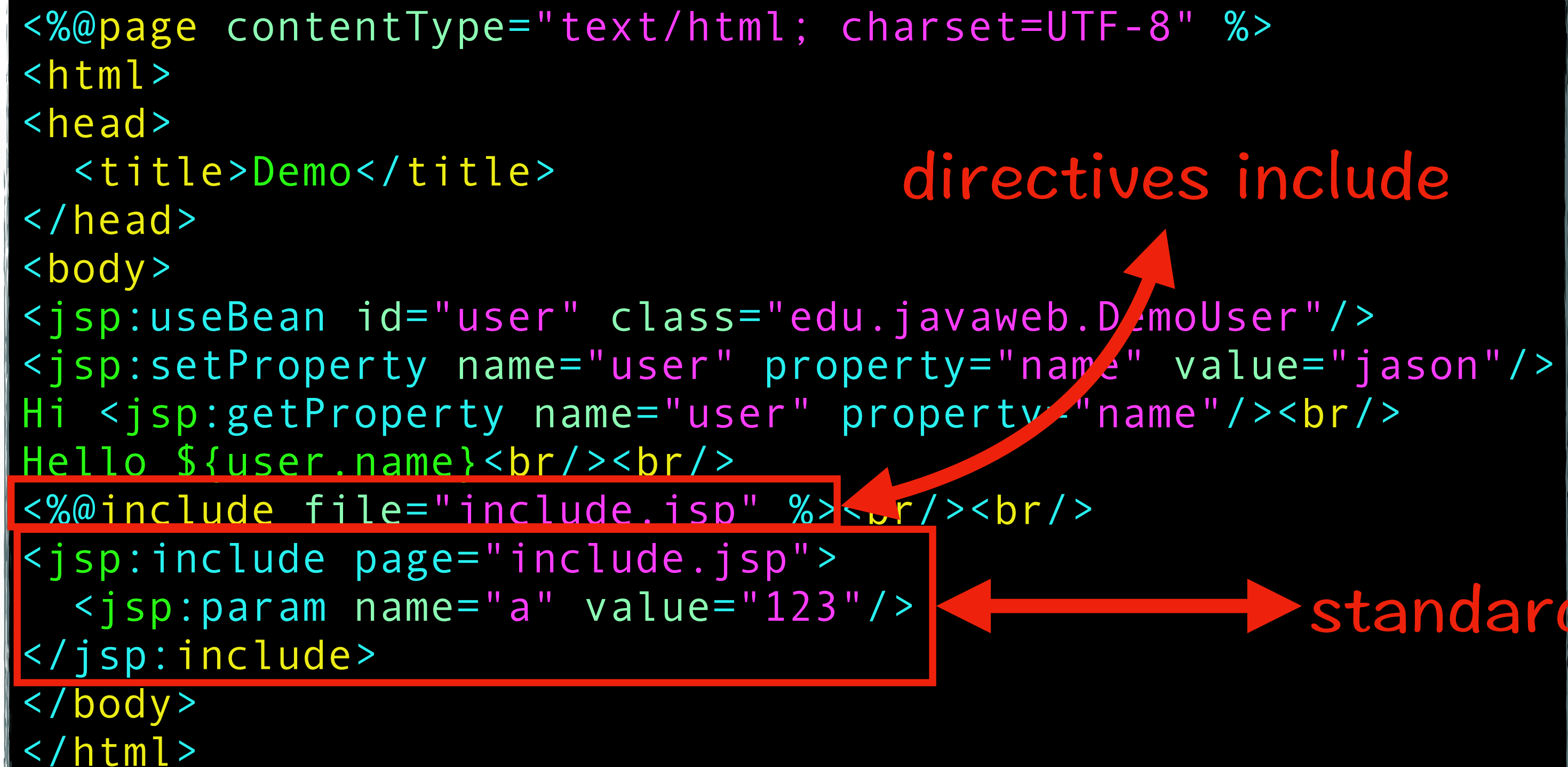
<jsp:include>	<%@include %>
屬性page	屬性file
page屬性接受相對和絕對URL路徑	file屬性接受相對和絕對URL路徑
在request期間包含目標頁面	目標檔案在轉譯階段加入
目標頁面可以包含動態程式碼	目標頁面可以包含動態程式碼
可有邏輯條件判斷是否加入	無法加入條件；無法嵌入頁面邏輯
收到request之前目標頁面可不存在	轉譯時目標檔案需存在
總是包含目標頁面最新版本	不一定包含目標檔案最新版：取決於容器(JSP未規範)

JSP standard action

include

- standard action include
- directives include

```
<%@page contentType="text/html; charset=UTF-8" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <jsp:useBean id="user" class="edu.javaweb.DemoUser"/>
  <jsp:setProperty name="user" property="name" value="jason"/>
  Hi <jsp:getProperty name="user" property="name"/><br/>
  Hello ${user.name}<br/><br/>
  <%@include file="include.jsp" %><br/><br/>
  <jsp:include page="include.jsp">
    <jsp:param name="a" value="123"/>
  </jsp:include>
</body>
</html>
```



Java Web

應用與實務

- JSP tag library
- 安裝JSTL
- 使用JSTL

JSP架構

Template Text

Elements

Directives

page

include

taglib

Scripting

Language-Base

declaration

scriptlet

expression

comment

EL

Action

Standard

Custom

JSTL

Tag File

Simple Tag

Classic Tag

Java Web

JSP tag library

- 下載JSTL 3.0
 - 規格
 - <https://mvnrepository.com/artifact/jakarta.servlet.jsp.jstl/jakarta.servlet.jsp.jstl-api/3.0.0>
- 實作
 - <https://mvnrepository.com/artifact/org.glassfish.web/jakarta.servlet.jsp.jstl/3.0.1>

Java Web

JSP tag library

- 使用JSTL
 - core
 - format
 - function
 - sql
 - xml

JSP tag library

使用JSTL

- core
 - `<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>`
 - 流程處理
 - 錯誤處理
 - URL處理
 - 其他

使用JSTL

core

- 流程處理

- 判斷式

- <c:if> : 單判斷

- <c:choose> : 多判斷

- <c:when> : 單一判斷

- <c:otherwise> : when都不成立

條件成立執行標籤本體

```
<%@page contentType="text/html" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <c:if test="${param.a>0}">
    Hi
  </c:if>
  <c:choose>
    <c:when test="${param.a>10}">OK</c:when>
    <c:when test="${param.a<10}">NO</c:when>
    <c:otherwise>XD</c:otherwise>
  </c:choose>
</body>
</html>
```

多條件判斷

<when>條件成立執行，執行後跳出<choose>

<when>條件都不成立執行<otherwise>

使用JSTL

core

- 流程處理

- 迴圈

```
var entities = map.entities();
```

```
var status = 0;
```

```
for(var e: entities){
```

```
    out.println(e.getKey()+"="+
```

```
        e.getValue());
```

```
    out.println(status++);
```

```
}
```

```
var delims = ":";
```

```
for(var i:"D:E:F".split(delims)){
```

```
    out.print(i);
```

```
}
```

```
<%@page contentType="text/html" %>
<%@page import="java.util.*" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
    <title>Demo</title>
</head>
<body>
    <c:forEach var="i" begin="0" end="10" step="2">
        ${i}<br/>
    </c:forEach>
    <%
        pageContext.setAttribute("list", List.of("A","B","C"));
        pageContext.setAttribute("map", Map.of("A",1,"B",2));
    %>
    <c:forEach var="i" items="${list}">
        ${i}<br/>
    </c:forEach>
    <c:forEach var="item" items="${map}" varStatus="status">
        ${item.key}=${item.value}<br/>
        ${status.index}<br/>
    </c:forEach>
    <c:forEach var="i" items="A,B,C">
        ${i}
    </c:forEach>
    <c:forTokens var="i" delims=":" items="D:E:F">
        ${i}
    </c:forTokens>
</body>
</html>
```

```
for(var i=0;i<=10;i+=2){
    out.println(i);
}
```

```
for(var i: list){
    out.println(i);
}
```

```
for(var i:"A,B,C".split(",")){
    out.print(i);
}
```

使用JSTL

core

- 流程處理
 - 迴圈的状态

屬性	說明
index	回傳目前物件是第幾個，由0開始
count	回傳已走過多少物件，由1開始
first	回傳目前物件是否為第一個
last	回傳目前物件是否為最後一個
current	回傳目前物件
begin	回傳begin值
end	回傳end值
step	回傳step值

使用JSTL

core

- 錯誤處理
- 將發生的錯誤處理掉將不進入error page
- 將可能發生錯誤的部份放置body區塊，使用var設定exception的參考變數名

判斷例外物件做處理

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
    <title>JSTL c:catch Example</title>
</head>
<body>
    <c:catch var="exception">
        <%
            int result = 10 / 0;
            pageContext.setAttribute("result", result);
        %>
    </c:catch>
    <h3>Division Result:</h3>
    <c:choose>
        <c:when test="${not empty exception}">
            <p>Error: Division by zero is not allowed.</p>
        </c:when>
        <c:otherwise>
            <p>Result: ${result}</p>
        </c:otherwise>
    </c:choose>
</body>
</html>
```

捕捉例外

使用JSTL

core

- URL處理
 - `<c:import url=" ">`
 - url指向一個有效位址(可非本機)，類似`<jsp:include>`
 - `<c:redirect url=" ">`
 - url指向一個有效位址(可非本機)
 - `<c:url value=" ">`
 - value為url位址，可輸出`encodeURL`，使無`cookie`環境亦可使用`session`
 - `<c:param name=" " value=" ">`
 - 可於`url`, `import`及`redirect`等標籤中設定`param`

使用JSTL

core

- URL處理

包含另一個網頁

```
<%@page contentType="text/html" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <c:import url="include.jsp">
    <c:param name="a" value="Hi"/>
  </c:import>
</body>
</html>
```

URLEncode的鏈結

```
<%@page contentType="text/html" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <a href="<c:url value='include.jsp' />">demo</a>
</body>
</html>
```

重導向到另一個網頁

```
<%@page contentType="text/html" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <c:redirect url="https://www.google.com">
    <c:param name="q" value="java"/>
  </c:redirect>
</body>
</html>
```


使用JSTL

core

- 其他

- `<c:set var=" " scope=" " target=" " property=" " value=" "/>`

- 設定屬性值
- var為變數名，target為物件或map名；不可一起使用
- scope值為page, request, session, application

- `<c:remove var=" " scope=" "/>`

- 移除屬性值

- `<c:out value=" " escapeXml="true" default=" "/>`

- 一般輸出，escapeXml為true則會轉譯內容裡所含的html，使其正常顯示；預設為true
- value為null時顯示default

使用JSTL

core

設定物件屬性

輸出物件屬性

```
<%@page contentType="text/html" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <jsp:useBean id="user" class="edu.javaweb.DemoUser"/>
  <c:set target="${user}" property="name" value="${param.name}"/>
  <c:set target="${user}" property="age" value="18"/>
  name: <c:out value="${user.name}" default="jason"/><br/>
  age: <c:out value="${user.age}"/><br/>
  <c:set var="a" value="1"/>
  <c:set var="b" value="2"/>
  <c:remove var="b"/>
  <c:out value="${a + b}"/>
</body>
</html>
```

移除變數值

設定變數值

JSP tag library

使用JSTL

- format
 - `<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>`
- 整頁設定
 - `<setLocale>`, `<setBundle>`, `<setTimeZone>`, `<requestEncoding>`
- 區域設定
 - `<bundle>`, `<timeZone>`
- i18n
 - `<formatNumber>`, `<parseNumber>`, `<formatDate>`, `<parseDate>`, `<message>`, `<param>`

使用JSTL

format

- format

/WEB-INF/classes/message.properties

```
foo.bar=好的  
foo.bar1={0} 是一個 {1}
```

/WEB-INF/classes/message_en_US.properties

```
foo.bar=OK  
foo.bar1={0} is a {1}
```

建立資源檔

```
<%@page contentType="text/html; charset=UTF-8" %>  
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>  
<html>  
<head>  
  <title>Demo</title>  
</head>  
<body>  
  <fmt:setLocale value="${param.locale}" />  
  <fmt:bundle basename="message">  
    <fmt:message key="foo.bar" /><br />  
    <fmt:message key="foo.bar1">  
      <fmt:param value="Apple" />  
      <fmt:param value="Fruit" />  
    </fmt:message>  
  </fmt:bundle>  
</body>  
</html>
```

設定語系

取得資源檔內容

使用JSTL

format

- format

```
<%@page contentType="text/html; charset=UTF-8" %>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <fmt:setLocale value="${param.locale}" />
  <jsp:useBean id="now" class="java.util.Date" />
  <fmt:formatDate value="${now}" type="date" dateStyle="full" /><br />
  <fmt:formatDate value="${now}" type="both" pattern="yyyy-MM-dd HH:mm:ss" /><br />
  <fmt:timeZone value="GMT+0">
    <fmt:formatDate value="${now}" type="time" dateStyle="full" /><br />
  </fmt:timeZone>
  <br />
  <fmt:formatNumber value="12345.678" /><br />
  <fmt:formatNumber value="12345.678" type="currency" /><br />
  <fmt:formatNumber value="12345.678" type="currency" currencySymbol="Sun" /><br />
  <fmt:formatNumber value="12345.678" type="percent" /><br />
</body>
</html>
```

設定語系

時間格式

時區

貨幣及數字格式

JSP tag library

使用JSTL

- function
 - `<%@taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/function" %>`
- EL函式庫
 - 字串大小寫: `toLowerCase`, `toUpperCase`
 - 取得子字串: `substring`, `substringAfter`, `substringBefore`
 - 去除前後空白: `trim`
 - 字串取代: `replace`, `escapeXML`
 - 檢查字串: `startsWith`, `endsWith`, `contains`, `containsIgnoreCase`
 - 字串位置: `indexOf`
 - 分割與連接字串: `split`, `join`
 - 取得長度: `length`

使用JSTL

function

- function

```
<%@page contentType="text/html" %>
<%@taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  ${fn:toUpperCase("apple")}<br/>
  ${fn:indexOf("apple is a fruit", "is")}<br/>
  ${fn:trim(fn:substring("apple is a fruit", 0, 9))}<br/>
  ${fn:substringAfter("apple is a fruit", "is")}<br/>
  ${fn:substringBefore("apple is a fruit", "is")}<br/>
  ${fn:length(fn:split("apple is a fruit", " "))}<br/>
</body>
</html>
```


JSP tag library

使用JSTL

- sql
 - `<%@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>`
 - 設定連線
 - `<setDataSource>`
 - 執行SQL
 - `<query>`, `<update>`, `<param>`, `<dateParam>`
 - 啟動交易
 - `<transaction>`

使用JSTL

sql

設定資料庫連線

新增資料

修改資料

刪除資料

詢查單筆資料

詢查所有資料

```
<%@page contentType="text/html;charset=UTF-8" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
<html>
<head>
<title>Demo</title>
</head>
<body>
<sql:setDataSource driver='org.postgresql.Driver'
    user='postgres' password='123456' url='jdbc:postgresql://localhost:5432/web'/>
<c:if test='${param.action} == "create" and not empty param.id and not empty param.name'>
<sql:update>
    insert into foo(id, name) values (?,?)
    <sql:param value='${param.id}'/>
    <sql:param value='${param.name}'/>
</sql:update>
</c:if>
<c:if test='${param.action} == "update" and not empty param.id and not empty param.name'>
<sql:update>
    update foo set name=? where id=?
    <sql:param value='${param.name}'/>
    <sql:param value='${param.id}'/>
</sql:update>
</c:if>
<c:if test='${param.action} == "delete" and not empty param.id'>
<sql:update>
    delete from foo where id=?
    <sql:param value='${param.id}'/>
</sql:update>
</c:if>
```

```
<c:choose>
<c:when test="${param.action} == 'query' and not empty param.id">
<sql:query var="foo">
    select id,name from foo where id = ?
    <sql:param value='${param.id}'/>
</sql:query>
<c:forEach var='f' items='${foo.rows}'>
    ${f.id} = ${f.name}<br/>
</c:forEach>
</c:when>
<c:otherwise>
<sql:query sql="select id,name from foo" var="foo"/>
<c:forEach var='f' items='${foo.rows}'>
    ${f.id} = ${f.name}<br/>
</c:forEach>
</c:otherwise>
</c:choose>
</body>
</html>
```

JSP tag library

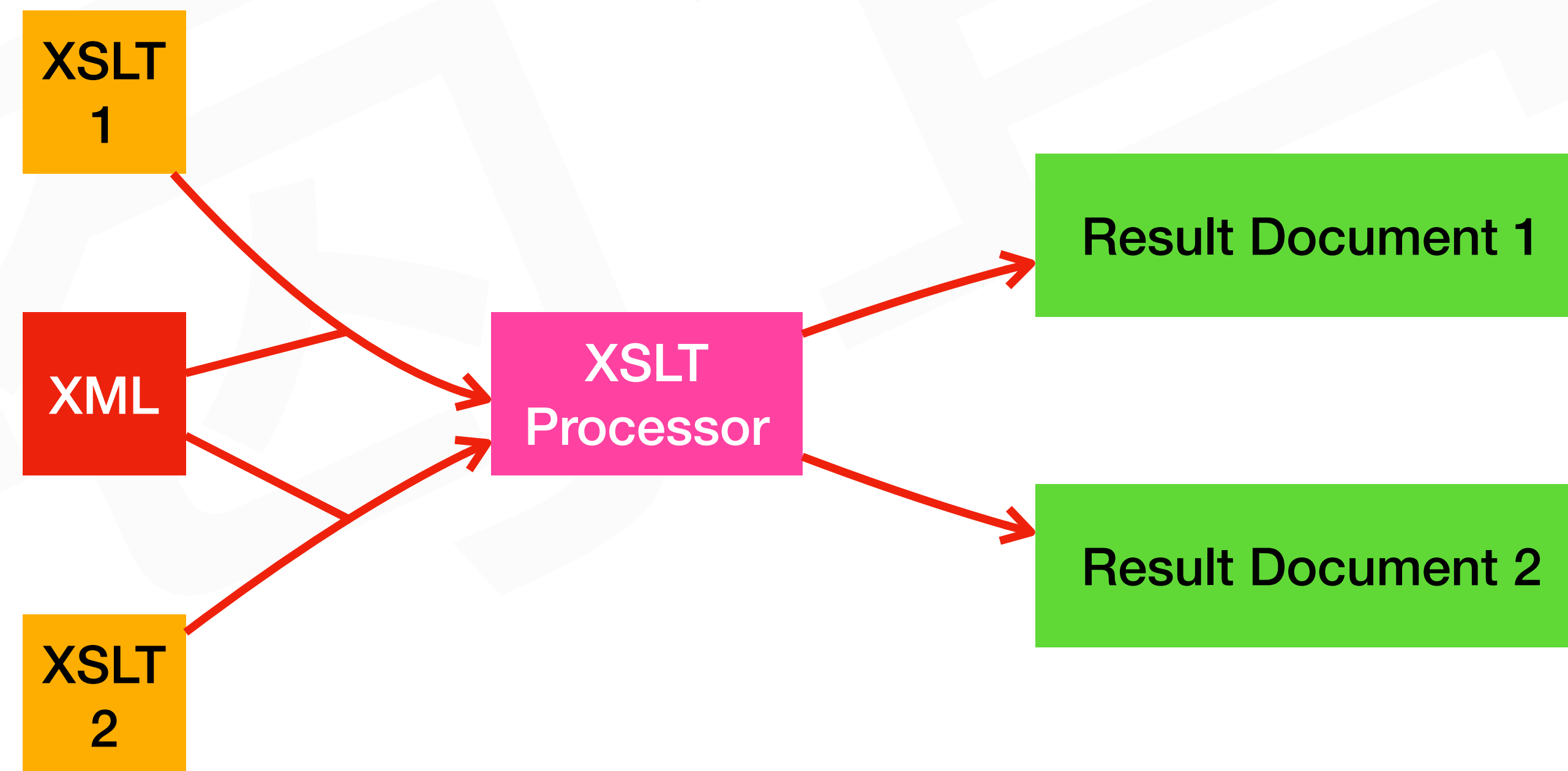
使用JSTL

- xml
 - XSLT
 - XPath
 - xalan

使用JSTL

xml

- XSLT
 - 將一份XML經過XSL轉換成另一個格式



使用JSTL

xml

- XPath
 - 用來查詢XML內容的語法

```
<bookstore>
  <book category="fiction">
    <title lang="en">Harry Potter</title>
    <author>J.K. Rowling</author>
    <price>29.99</price>
  </book>
  <book category="technical">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <price>39.95</price>
  </book>
</bookstore>
```

坐標	表示式	說明
/	/bookstore	取得節點
//	//book	取得所有book節點
.		當前節點
..		父節點
@	//book[@category='fiction']	取得節點的屬性
[1]	/bookstore/book[1]	取第1個節點
[last()]	/bookstore/book[last()]	取最後1個節點
[position()<3]	/bookstore/book[position()<3]	取得前2個節點
*	/bookstore/*	取得子節點

使用JSTL

xml

- xalan
 - JSTL 1.2
 - xalan:實作XSLT及XPath
 - <http://ftp.tc.edu.tw/pub/Apache/xalan/xalan-j/binaries/>
 - 將jar放置於<CATALINA_HOME>/lib
 - JSTL 3.0已包含在套件中

使用JSTL

xml

- `<%@taglib prefix='x' uri='http://java.sun.com/jsp/jstl/xml' %>`
- 輸入輸出參數
 - `<out>`, `<parse>`, `<set>`
- 流程控制
 - `<forEach>`, `<if>`, `<choose>`, `<when>`, `<otherwise>`
- XSLT
 - `<transform>`, `<param>`
 - `<transform>` 將XSL轉換XML
 - `<param>` 設置XSL樣式表

使用JSTL

xml

請取XML資料

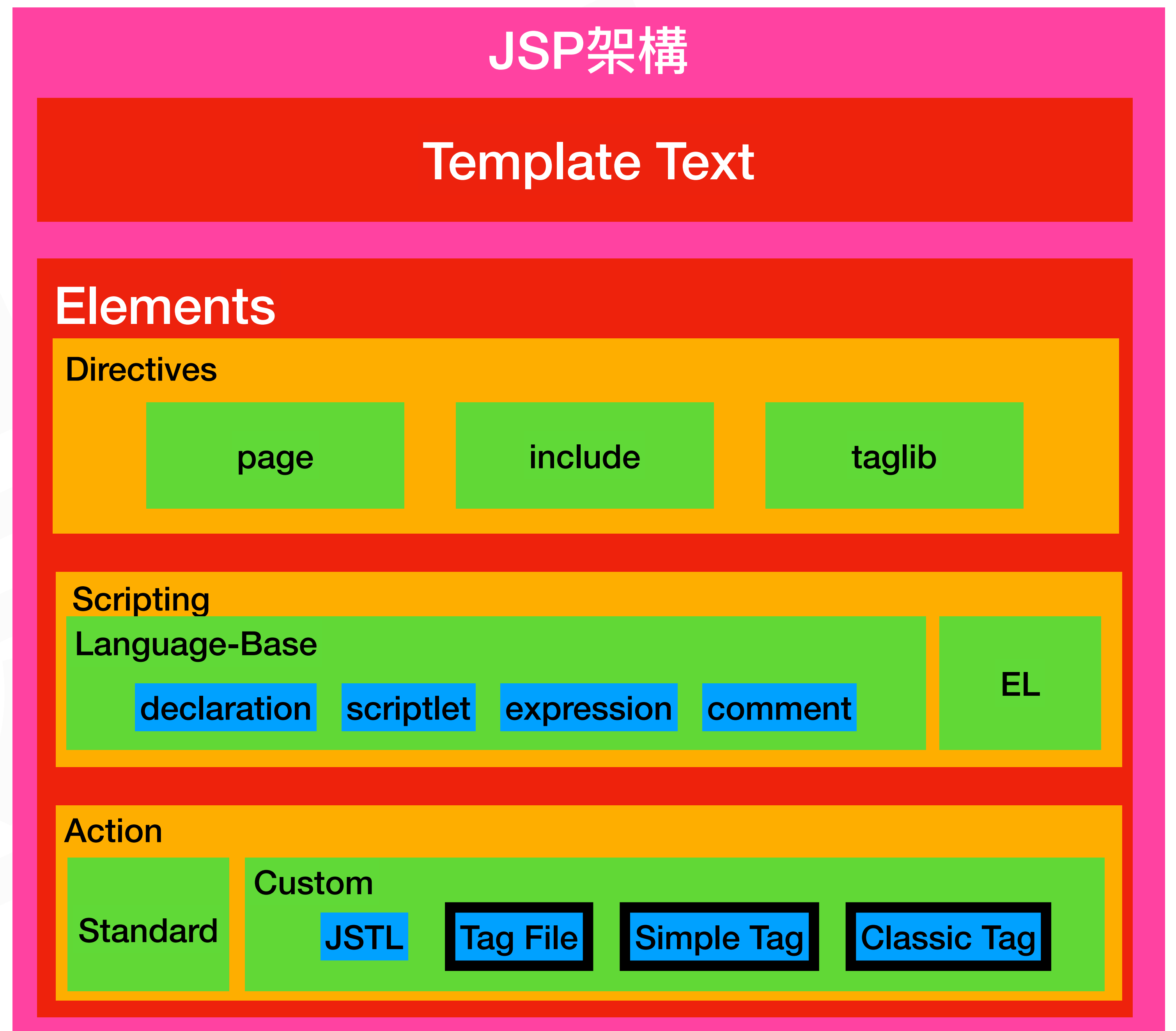
```
<%@page contentType="text/html; charset=UTF-8" %>
<%@taglib prefix='x' uri='http://java.sun.com/jsp/jstl/xml' %>
<html>
<head>
  <meta charset='utf-8' />
  <title>Demo</title>
</head>
<body>
  <x:parse var="users">
    <users>
      <user id='1'>
        <name>Jason</name>
        <age>18</age>
      </user>
      <user id='2'>
        <name>John</name>
        <age>20</age>
      </user>
    </users>
  </x:parse>
  <table>
    <thead>
      <tr>
        <td>id</td>
        <td>name</td>
        <td>age</td>
      </tr>
    </thead>
    <tbody>
      <x:forEach var="user" select="$users//user">
        <tr>
          <td><x:out select="$user/@id"/></td>
          <td><x:out select="$user/name"/></td>
          <td><x:out select="$user/age"/></td>
        </tr>
      </x:forEach>
    </tbody>
  </table>
</body>
</html>
```

輸出XML資料

Java Web

應用與實務

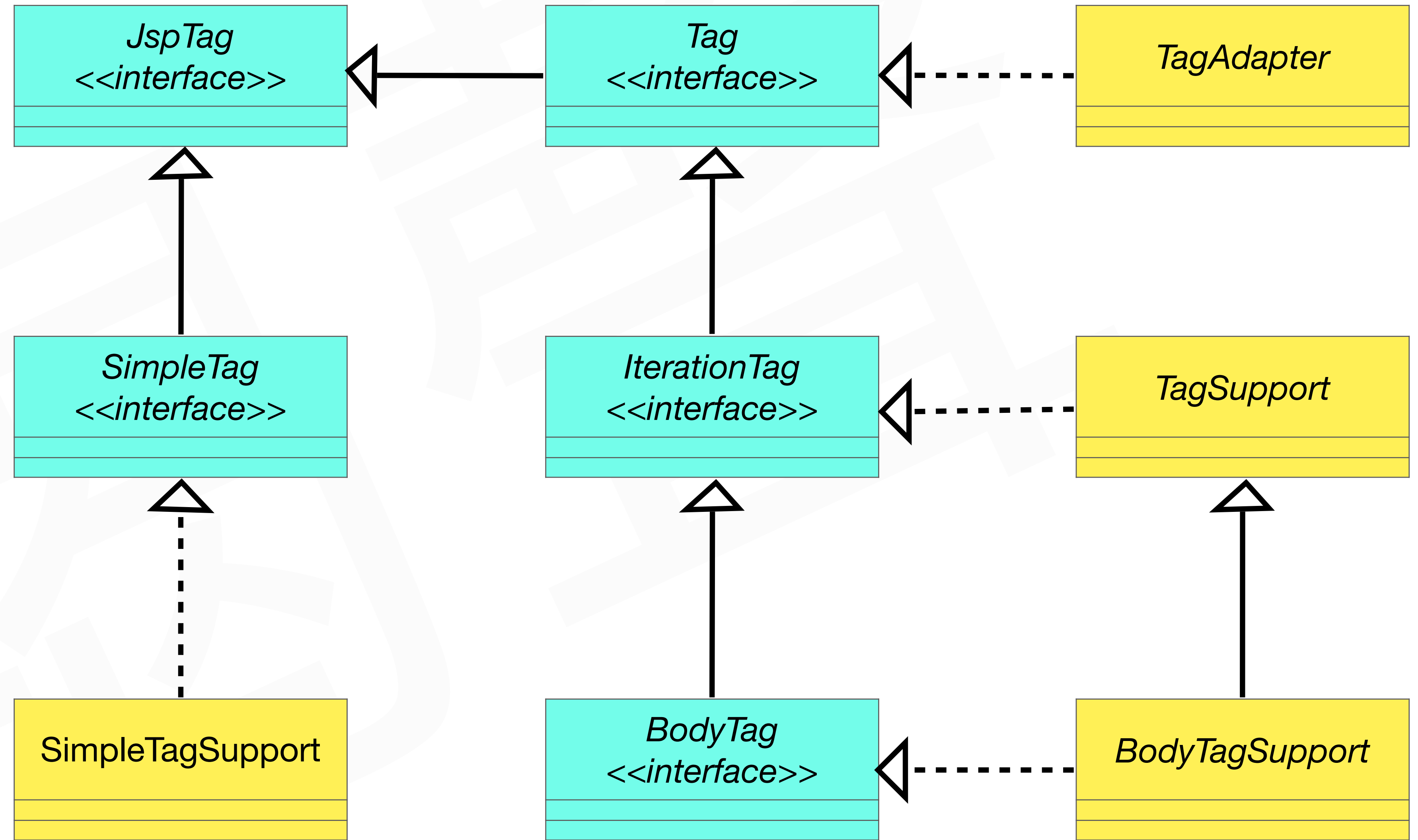
- Custom Tags
 - Tag File
 - Simple Tag
 - Classic Tag



Java Web

應用與實務

- JSP Tags



Java Web

Custom Tags

- Tag File
 - 不需要寫程式
 - 放置在/WEB-INF/tags或子目錄中
 - 打包成jar使用tld來指定位置
- 轉譯後繼承SimpleTagSupport

Custom Tags

Tag File

```
<%@page contentType="text/html;charset=UTF-8" %>
<%@taglib prefix='html' tagdir='/WEB-INF/tags' %>
<html>
<html:header title='Demo'>
  <meta charset='utf-8' />
</html:header>
<body>
OK
</body>
</html>
```

標籤屬性

檔案所在目錄

標籤名為檔名

`/WEB-INF/tags/header.tag`

```
<%@tag description='header' pageEncoding='UTF-8' %>
<%@tag body content='scriptless' %>
<%@attribute name='title' %>
<head>
  <title>${title}</title>
  <jsp:doBody/>
</head>
```


Custom Tags

Tag File

本體內容

scriptless(不可出現<% %>, <%! %>, <%= %>)

empty(標籤無body)

tagdependent(將body為純文字處理)

/WEB-INF/tags/header.tag

```
<%@tag description='header' pageEncoding='UTF-8' %>
<%@tag body-content='scriptless' %>
<%@attribute name='title' %>
<head>
  <title>${title}</title>
  <jsp:doBody/>
</head>
```

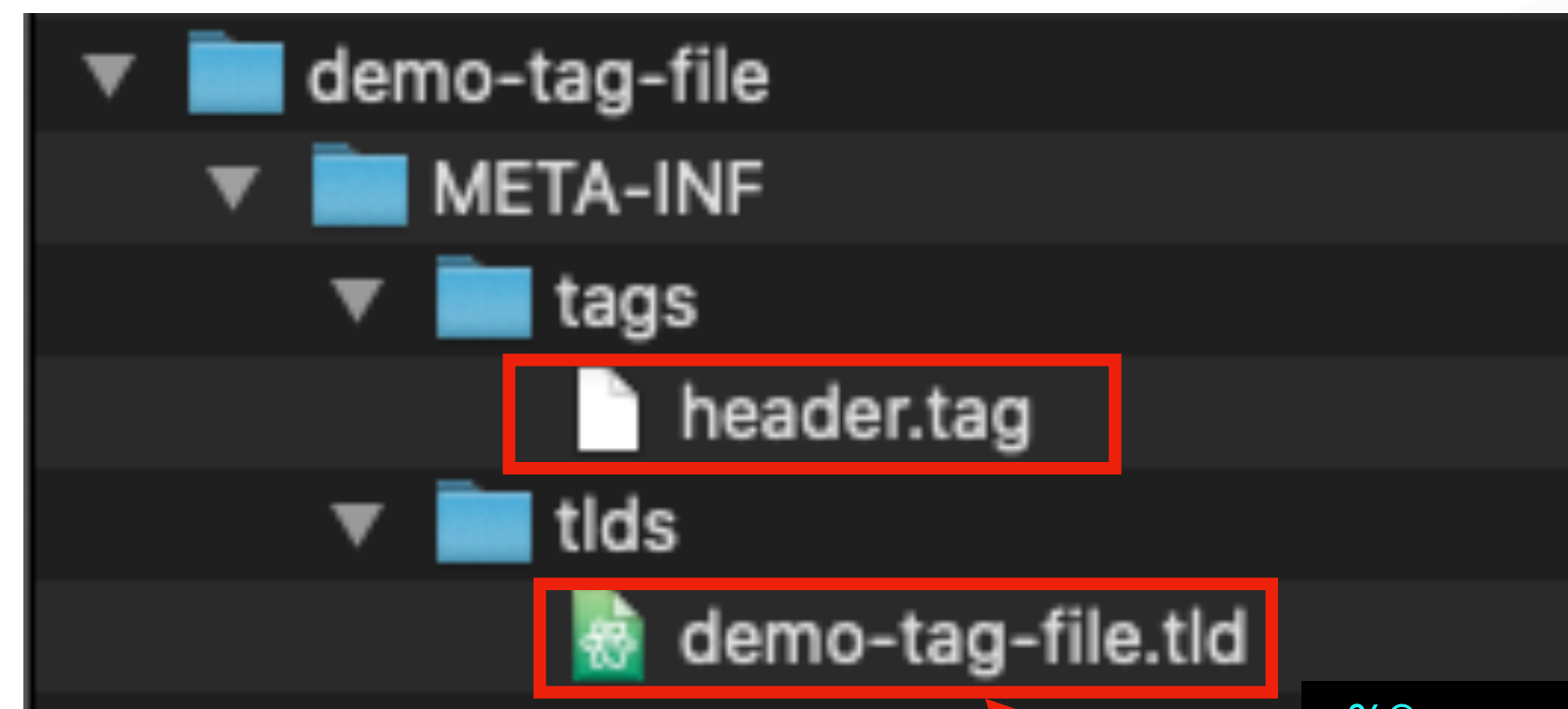
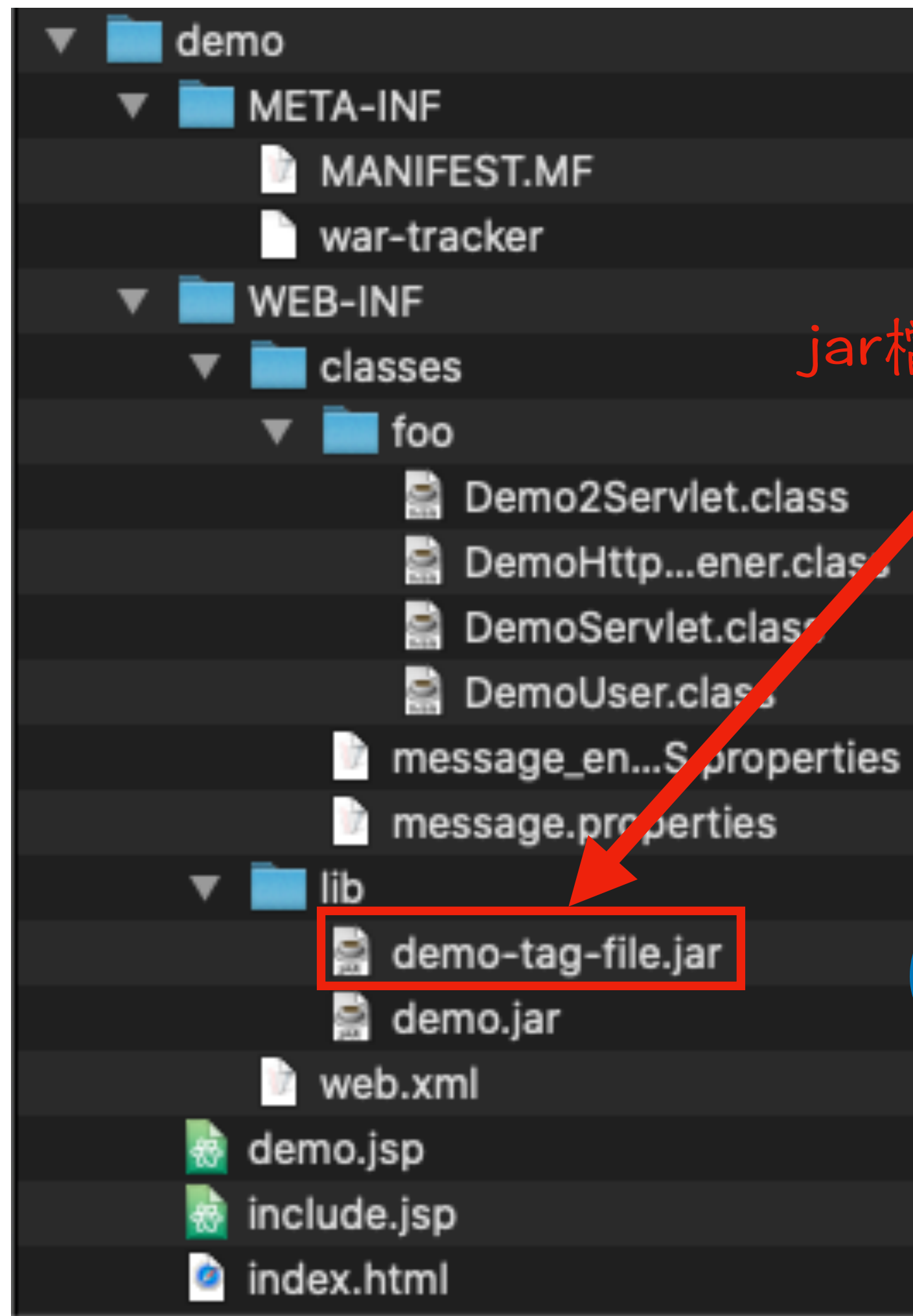
取得屬性值

標籤本體

```
<%@page contentType="text/html; charset=UTF-8" %>
<%@taglib prefix='html' tagdir='/WEB-INF/tags' %>
<html>
<html:header title='Demo'>
  <meta charset='utf-8' />
</html:header>
<body>
OK
</body>
</html>
```

Custom Tags

Tag File



```
<%@page contentType="text/html; charset=UTF-8" %>
<%@taglib prefix='html' uri='http://demo.tag/header' %>
<html>
<html:header title='Demo'>
  <meta charset='utf-8' />
</html:header>
<body>
  OK
</body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
  version="2.1">
  <tlib-version>1.0</tlib-version>
  <short-name>demoTagFile</short-name>
  <uri>http://demo.tag/header</uri>
  <tag-file>
    <name>header</name>
    <path>/META-INF/tags/header.tag</path>
  </tag-file>
</taglib>
```

標籤檔案設定

Java Web

Custom Tags

- Simple Tag
 - 提供較為簡單的自定標籤方式
 - 不可執行太複雜的動作
 - 每次產生新的物件

Custom Tags

Simple Tag

- 簡單標籤處理器流程
 - 建立自訂標籤處理器實例
 - 呼叫標籤處理器的setJspContext()方法設定PageContext實例
 - 如果是巢狀標籤中的內層標籤，還會呼叫標籤處理器的setParent()方法，並傳入外層標籤處理器的實例
 - 設定標籤處理器的屬性(setXxx)
 - 呼叫標籤處理器的setJspBody()方法設定JspFragment實例
 - 如果有body才會呼叫
 - 呼叫標籤處理器的doTag()方法
 - 處理網頁剩餘部份
 - 如果SkipPageException丟出來時，不處理剩餘部份
- 銷毀標籤處理器實例

Custom Tags

Simple Tag

- 標籤本體
 - 建立標籤處理器，繼承SimpleTagSupport
 - Override doTag()
 - 取得標籤本體(JspFragment)使用getJspBody()
 - 執行標籤本體：使用JspFragment的invoke(JspWriter)
 - invoke(null)使用預設方式(JspWriter)輸出
 - 若有傳入物件(如StringWriter)，則由傳入物件取得標籤本體區塊內容，再使用getJspContext().getOut()取得原輸出物件
 - invoke()執行一次，body區塊就會輸出一次；重覆輸出可用迴圈

Custom Tags

Simple Tag

- 標籤描述檔
 - body-content：值為scriptless, empty, tagdependent
 - 特性與Tag File相同
- attribute
 - name：屬性名稱
 - required：是否必須
 - rtexprvalue：是否可使用runtime expression value(EL)
 - type：屬性型態

Custom Tags

Simple Tag

```
<%@page contentType="text/html;charset=UTF-8" %>
<%@taglib prefix='z' uri='http://foo.demo/jstl/z' %>
<html>
<head>
  <meta charset='utf-8' />
  <title>Demo</title>
</head>
<body>
<z:if test="${param.a>10}">
  Hi
</z:if>
<z:choose>
  <z:when test="${param.a>10}">OK</z:when>
  <z:when test="${param.a<10}">NO</z:when>
  <z:otherwise>XD</z:otherwise>
</z:choose>
</body>
</html>
```

需對應

標籤名稱及屬性對應

```
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
  version="2.1">
  <tlib-version>1.0</tlib-version>
  <short-name>z</short-name>
  <uri>http://foo.demo/jstl/z</uri>
  <tag>
    <name>if</name>
    <tag-class>edu.javaweb.DemoIfSimpleTag</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
      <name>test</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>boolean</type>
    </attribute>
  </tag>
  <tag>
    <name>choose</name>
    <tag-class>edu.javaweb.DemoChooseSimpleTag</tag-class>
    <body-content>scriptless</body-content>
  </tag>
  <tag>
    <name>when</name>
    <tag-class>edu.javaweb.DemoWhenSimpleTag</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
      <name>test</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>boolean</type>
    </attribute>
  </tag>
  <tag>
    <name>otherwise</name>
    <tag-class>edu.javaweb.DemoOtherwiseSimpleTag</tag-class>
    <body-content>scriptless</body-content>
  </tag>
</taglib>
```

Custom Tags

Simple Tag

```
package edu.javaweb;
import java.io.*;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoIfSimpleTag extends SimpleTagSupport{
    private boolean test;
    public void doTag() throws JspException{
        if(test){
            try{
                getJspBody().invoke(null);
            }catch(IOException e){
                throw new JspException("error", e);
            }
        }
    }
    public void setTest(boolean test){
        this.test = test;
    }
}
```

處理標籤本體

```
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
version="2.1">
<tlib-version>1.0</tlib-version>
<short-name>z</short-name>
<uri>http://foo.demo/istl/z</uri>
<tag>
    <name>if</name>
    <tag-class>edu.javaweb.DemoIfSimpleTag</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
        <name>test</name>
        <required>true</required>
        <rtexprvalue>true</rtexprvalue>
        <type>boolean</type>
    </attribute>
</tag>
<tag>
    <name>choose</name>
    <tag-class>edu.javaweb.DemoChooseSimpleTag</tag-class>
    <body-content>scriptless</body-content>
</tag>
<tag>
    <name>when</name>
    <tag-class>edu.javaweb.DemoWhenSimpleTag</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
        <name>test</name>
        <required>true</required>
        <rtexprvalue>true</rtexprvalue>
        <type>boolean</type>
    </attribute>
</tag>
<tag>
    <name>otherwise</name>
    <tag-class>edu.javaweb.DemoOtherwiseSimpleTag</tag-class>
    <body-content>scriptless</body-content>
</tag>
</taglib>
```

Custom Tags

Simple Tag

```
package edu.javaweb;
import java.io.*;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoChooseSimpleTag extends SimpleTagSupport{
    private boolean matched;
    public boolean isMatched(){
        return this.matched;
    }
    public void setMatched(boolean matched){
        this.matched = matched;
    }
    public void doTag() throws JspException {
        try{
            this.getJspBody().invoke(null);
        }catch(IOException e){
            throw new JspException("choose error", e);
        }
    }
}
```

```
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
            http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
        version="2.1">
    <tlib-version>1.0</tlib-version>
    <short-name>z</short-name>
    <uri>http://foo.demo/jstl/z</uri>
    <tag>
        <name>if</name>
        <tag-class>edu.javaweb.DemoIfSimpleTag</tag-class>
        <body-content>scriptless</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>choose</name>
        <tag-class>edu.javaweb.DemoChooseSimpleTag</tag-class>
        <body-content>scriptless</body-content>
    </tag>
    <tag>
        <name>when</name>
        <tag-class>edu.javaweb.DemoWhenSimpleTag</tag-class>
        <body-content>scriptless</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>otherwise</name>
        <tag-class>edu.javaweb.DemoOtherwiseSimpleTag</tag-class>
        <body-content>scriptless</body-content>
    </tag>
</taglib>
```


Custom Tags

Simple Tag

```
package edu.javaweb;
import java.io.*;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoWhenSimpleTag extends SimpleTagSupport{
    private boolean test;
    public void setTest(boolean test){
        this.test = test;
    }
    public void doTag() throws JspException{
        JspTag parent = null;
        if(!((parent = getParent()) instanceof DemoChooseSimpleTag)){
            throw new JspTagException("Parent isn't DemoChooseSimpleTag");
        }
        if(((DemoChooseSimpleTag)parent).isMatched()){
            return;
        }
        if(test){
            ((DemoChooseSimpleTag)parent).setMatched(true);
            try{
                this.getJspBody().invoke(null);
            }catch(IOException e){
                throw new JspException("when error", e);
            }
        }
    }
}
```

```
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
version="2.1">
<tlib-version>1.0</tlib-version>
<short-name>z</short-name>
<uri>http://foo.demo/jstl/z</uri>
<tag>
    <name>if</name>
    <tag-class>edu.javaweb.DemoIfSimpleTag</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
        <name>test</name>
        <required>true</required>
        <rtexprvalue>true</rtexprvalue>
        <type>boolean</type>
    </attribute>
</tag>
<tag>
    <name>choose</name>
    <tag-class>edu.javaweb.DemoChooseSimpleTag</tag-class>
    <body-content>scriptless</body-content>
</tag>
<tag>
    <name>when</name>
    <tag-class>edu.javaweb.DemoWhenSimpleTag</tag-class>
    <body-content>scriptless</body-content>
    <attribute>
        <name>test</name>
        <required>true</required>
        <rtexprvalue>true</rtexprvalue>
        <type>boolean</type>
    </attribute>
</tag>
<tag>
    <name>otherwise</name>
    <tag-class>edu.javaweb.DemoOtherwiseSimpleTag</tag-class>
    <body-content>scriptless</body-content>
</tag>
</taglib>
```

Custom Tags

Simple Tag

```
package edu.javaweb;
import java.io.*;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoOtherwiseSimpleTag extends SimpleTagSupport{
    public void doTag() throws JspException{
        JspTag parent = null;
        if(!((parent = getParent()) instanceof DemoChooseSimpleTag)){
            throw new JspTagException("Parent isn't DemoChooseSimpleTag");
        }
        if(((DemoChooseSimpleTag)parent).isMatched()){
            return;
        }
        try{
            this.getJspBody().invoke(null);
        }catch(IOException e){
            throw new JspException("otherwise error", e);
        }
    }
}
```

```
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
            http://java.sun.com/xml/ns/javaee/web-jstl/library_2_1.xsd"
        version="2.1">
    <tlib-version>1.0</tlib-version>
    <short-name>z</short-name>
    <uri>http://foo.demo/jstl/z</uri>
    <tag>
        <name>if</name>
        <tag-class>edu.javaweb.DemoIfSimpleTag</tag-class>
        <body-content>scriptless</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>choose</name>
        <tag-class>edu.javaweb.DemoChooseSimpleTag</tag-class>
        <body-content>scriptless</body-content>
    </tag>
    <tag>
        <name>when</name>
        <tag-class>edu.javaweb.DemoWhenSimpleTag</tag-class>
        <body-content>scriptless</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>otherwise</name>
        <tag-class>edu.javaweb.DemoOtherwiseSimpleTag</tag-class>
        <body-content>scriptless</body-content>
    </tag>
</taglib>
```

Java Web

Custom Tags

- Classic Tag
 - 實作較複雜
 - 可執行較複雜的動作
 - 不會每次產生新物件(具標籤池)

Custom Tags

Classic Tag

- 古典標籤處理器流程
 - 嘗試從Tag Pool找到可用的tag物件
 - 如果找到就直接使用
 - 如果找不到就建立新的tag物件
 - 呼叫標籤處理器的setPageContext()方法設定PageContext實例
 - 如果是巢狀標籤中的內層標籤，則還會呼叫標籤處理器的setParent()方法，並傳入外層標籤處理器的實例
 - 設定標籤處理器屬性(setXxx)
 - 呼叫標籤處理器的doStartTag()方法，並依不同的傳回值決定是否執行本體或呼叫doAfterBody()、doEndTag()方法
 - 將標籤處理器實例放入Tag Pool中以便再度使用

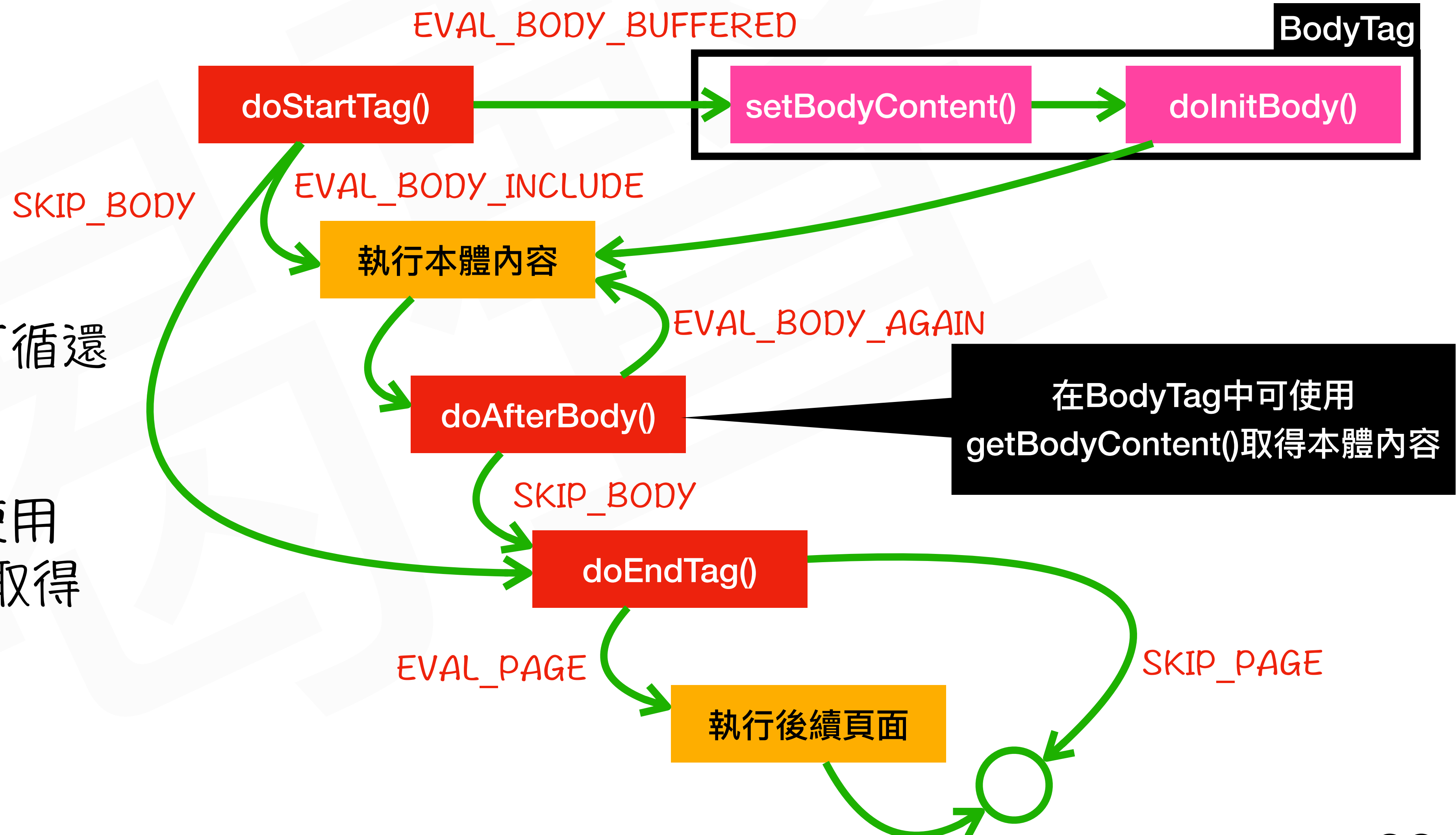
Custom Tags

Classic Tag

- 標籤本體

- IterationTag的doAfterBody()僅可循環還

- BodyTag的doAfterBody()可使用getBodyContent()取得本體內容



Custom Tags

Classic Tag

- 標籤描述檔
 - body-content：值為empty, JSP, tagdependent
 - JSP表示執行所有動態內容
- attribute
 - name：屬性名稱
 - required：是否必須
 - rtexprvalue：是否可使用runtime expression value
 - type：屬性型態

Custom Tags

Classic Tag

```
<%@page contentType="text/html;charset=UTF-8" %>
<%@taglib prefix="z" uri='http://foo.demo/jstl/z' %>
<%@taglib prefix="x" uri='http://foo.demo/jstl/x' %>
<html>
<head>
  <meta charset='utf-8' />
  <title>Demo</title>
</head>
<body>
<x:if test="${param.a>10}">
  Hi
</x:if>
<x:choose>
  <x:when test="${param.a>10}">OK</x:when>
  <x:when test="${param.a<10}">NO</x:when>
  <x:otherwise>XD</x:otherwise>
</x:choose>
</body>
</html>
```

需對應

標籤名稱及屬性對應

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
  version="2.1">
  <tlib-version>1.0</tlib-version>
  <short-name>x</short-name>
  <uri>http://foo.demo/jstl/x</uri>
  <tag>
    <name>if</name>
    <tag-class>edu.javaweb.DemoIfTag</tag-class>
    <body-content>JSP</body-content>
    <attribute>
      <name>test</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>boolean</type>
    </attribute>
  </tag>
  <tag>
    <name>choose</name>
    <tag-class>edu.javaweb.DemoChooseTag</tag-class>
    <body-content>JSP</body-content>
  </tag>
  <tag>
    <name>when</name>
    <tag-class>edu.javaweb.DemoWhenTag</tag-class>
    <body-content>JSP</body-content>
    <attribute>
      <name>test</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
      <type>boolean</type>
    </attribute>
  </tag>
  <tag>
    <name>otherwise</name>
    <tag-class>edu.javaweb.DemoOtherwiseTag</tag-class>
    <body-content>JSP</body-content>
  </tag>
</taglib>
```


Custom Tags

Classic Tag

```
package edu.javaweb;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoIfTag extends TagSupport{
    private boolean test;
    public int doStartTag() throws JspException{
        return test?EVAL_BODY_INCLUDE:SKIP_BODY;
    }
    public void setTest(boolean test){
        this.test = test;
    }
}
```

開始處理標籤

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
        version="2.1">
    <tlib-version>1.0</tlib-version>
    <short-name>x</short-name>
    <uri>http://foo.demo/istl/x</uri>
    <tag>
        <name>if</name>
        <tag-class>edu.javaweb.DemoIfTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>choose</name>
        <tag-class>edu.javaweb.DemoChooseTag</tag-class>
        <body-content>JSP</body-content>
    </tag>
    <tag>
        <name>when</name>
        <tag-class>edu.javaweb.DemoWhenTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>otherwise</name>
        <tag-class>edu.javaweb.DemoOtherwiseTag</tag-class>
        <body-content>JSP</body-content>
    </tag>
</taglib>
```

Custom Tags

Classic Tag

```
package edu.javaweb;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoChooseTag extends TagSupport{
    private boolean matched;
    public boolean isMatched(){
        return this.matched;
    }
    public void setMatched(boolean matched){
        this.matched = matched;
    }
    public int doStartTag() throws JspException{
        matched = false;
        return EVAL_BODY_INCLUDE;
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
        version="2.1">
    <tlib-version>1.0</tlib-version>
    <short-name>x</short-name>
    <uri>http://foo.demo/jstl/x</uri>
    <tag>
        <name>if</name>
        <tag-class>edu.javaweb.DemoIfTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>choose</name>
        <tag-class>edu.javaweb.DemoChooseTag</tag-class>
        <body-content>JSP</body-content>
    </tag>
    <tag>
        <name>when</name>
        <tag-class>edu.javaweb.DemoWhenTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>otherwise</name>
        <tag-class>edu.javaweb.DemoOtherwiseTag</tag-class>
        <body-content>JSP</body-content>
    </tag>
</taglib>
```


Custom Tags

Classic Tag

```
package edu.javaweb;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoWhenTag extends BodyTagSupport{
    private boolean test;
    public void setTest(boolean test){
        this.test = test;
    }
    public int doStartTag() throws JspException{
        JspTag parent = getParent();
        if(!(parent instanceof DemoChooseTag)){
            throw new JspTagException("when error");
        }
        if(((DemoChooseTag)parent).isMatched() || !test){
            return SKIP_BODY;
        } else {
            ((DemoChooseTag) parent).setMatched(true);
            return EVAL_BODY_INCLUDE;
        }
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
        version="2.1">
    <tlib-version>1.0</tlib-version>
    <short-name>x</short-name>
    <uri>http://foo.demo/jstl/x</uri>
    <tag>
        <name>if</name>
        <tag-class>edu.javaweb.DemoIfTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>choose</name>
        <tag-class>edu.javaweb.DemoChooseTag</tag-class>
        <body-content>JSP</body-content>
    </tag>
    <tag>
        <name>when</name>
        <tag-class>edu.javaweb.DemoWhenTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>otherwise</name>
        <tag-class>edu.javaweb.DemoOtherwiseTag</tag-class>
        <body-content>JSP</body-content>
    </tag>
</taglib>
```

Custom Tags

Classic Tag

```
package edu.javaweb;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoOtherwiseTag extends TagSupport{
    public int doStartTag() throws JspException{
        JspTag parent = getParent();
        if(!(parent instanceof DemoChooseTag)){
            throw new JspTagException("otherwise error");
        }
        if(((DemoChooseTag)parent).isMatched()){
            return SKIP_BODY;
        }else{
            return EVAL_BODY_INCLUDE;
        }
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"
        version="2.1">
    <tlib-version>1.0</tlib-version>
    <short-name>x</short-name>
    <uri>http://foo.demo/jstl/x</uri>
    <tag>
        <name>if</name>
        <tag-class>edu.javaweb.DemoIfTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>choose</name>
        <tag-class>edu.javaweb.DemoChooseTag</tag-class>
        <body-content>JSP</body-content>
    </tag>
    <tag>
        <name>when</name>
        <tag-class>edu.javaweb.DemoWhenTag</tag-class>
        <body-content>JSP</body-content>
        <attribute>
            <name>test</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
            <type>boolean</type>
        </attribute>
    </tag>
    <tag>
        <name>otherwise</name>
        <tag-class>edu.javaweb.DemoOtherwiseTag</tag-class>
        <body-content>JSP</body-content>
    </tag>
</taglib>
```



Custom Tags

Classic Tag

- TagAdapter
 - 簡單標籤及古典標籤混用時的相容

子 父	Tag	SimpleTag
Tag	O	O
SimpleTag	TagAdapter	O

Custom Tags

Classic Tag

- TagAdapter

```
<%@page contentType="text/html;charset=UTF-8" %>
<%@taglib prefix='z' uri='http://foo.demo/jstl/z' %>
<%@taglib prefix='x' uri='http://foo.demo/jstl/x' %>
<html>
<head>
  <meta charset='utf-8' />
  <title>Demo</title>
</head>
<body>
<x:if test="${param.a>10}">
  Hi
</x:if>
<z:choose>
  <x:when test="${param.a>10}">OK</x:when>
  <x:when test="${param.a<10}">NO</x:when>
  <x:otherwise>XD</x:otherwise>
</z:choose>
</body>
</html>
```

父標籤為簡單標籤

子標籤為古典標籤

```
package edu.javaweb;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoOtherwiseTag extends TagSupport{
    public int doStartTag() throws JspException{
        JspTag parent = getParent();
        if(parent instanceof TagAdapter){
            parent = ((TagAdapter)parent).getAdaptee();
        }
        if(parent instanceof DemoChooseTag){
            if(((DemoChooseTag)parent).isMatched()){
                return SKIP_BODY;
            }else{
                return EVAL_BODY_INCLUDE;
            }
        } else if(parent instanceof DemoChooseSimpleTag){
            if(((DemoChooseSimpleTag)parent).isMatched()){
                return SKIP_BODY;
            }else{
                return EVAL_BODY_INCLUDE;
            }
        } else {
            throw new JspTagException("otherwise error");
        }
    }
}
```

古典標籤實作中

判斷TagAdapter

```
package edu.javaweb;
import jakarta.servlet.jsp.*;
import jakarta.servlet.jsp.tagext.*;
public class DemoWhenTag extends BodyTagSupport{
    private boolean test;
    public void setTest(boolean test){
        this.test = test;
    }
    public int doStartTag() throws JspException{
        JspTag parent = getParent();
        if(parent instanceof TagAdapter){
            parent = ((TagAdapter)parent).getAdaptee();
        }
        if(parent instanceof DemoChooseTag){
            if(((DemoChooseTag)parent).isMatched() || !test){
                return SKIP_BODY;
            } else {
                ((DemoChooseTag) parent).setMatched(true);
                return EVAL_BODY_INCLUDE;
            }
        } else if(parent instanceof DemoChooseSimpleTag){
            if(((DemoChooseSimpleTag)parent).isMatched() || !test){
                return SKIP_BODY;
            } else {
                ((DemoChooseSimpleTag)parent).setMatched(true);
                return EVAL_BODY_INCLUDE;
            }
        } else {
            throw new JspTagException("when error");
        }
    }
}
```