

網路應用概論

JSP動態網頁程式基礎篇

(Part_IV_JavaBasic.pdf)(Part_V_JavaOOP.pdf)

(Part_VII_JSP.pdf)(Part_VIII_JavaServlet.pdf)

網頁資料傳送 - 客戶端

- 可利用FrontPage製作表單，包含text, textarea, checkbox, radio, drop-down-list(select)等，主要傳送設定為
 - 傳送方式：post
 - 控制檔名：FormTest.jsp
 - 各欄位名稱如name屬性，傳送之資料為value屬性內的值，其中course及sport允許多重選擇，textarea則傳送標籤內的字串資料

```
<form action="FormTest.jsp" method="post">
<input type="text" name="textfield" value="" ... />
<textarea name="textarea"></textarea>
<input type="checkbox" name="course" value=... />
<input name="degree" type="radio" value=... />
<select name="department">
  <option value=...>...</option>
</select>
<select name="sport" size="5" multiple>
  <option selected value=...>...</option>
</select>
</form>
```

網頁資料傳送 - 伺服器端

- 以開發工具如netbeans或記事本撰寫FormTest.jsp
 - JSP為一種內嵌式語言，與html語法混合使用，主要架構如下
<%
JSP語法區塊, 即為Java語言(詳見Java語法介紹)
%>
 - out.println(“輸出字串” + 變數或函式);
 - 輸出字串即為以html語法格式呈現在網頁上的內容
 - 變數或函式則為jsp程式中所定義，在此會自動轉成字串型態
 - 利用加號(+)將字串連接起來
 - 如要換行，以
作為輸出字串
 - request.getParameter(“fieldname”)
 - 接收客戶端對應之表單內各元件定義於value屬性所傳之資料
 - fieldname即為元件名稱
 - request.getParameterValues(“fieldname”)
 - 表示接受到的資料為陣列格式(即多重選擇之資料)
 - 註解符號(Comment Line) : //

```
<%  
out.println("<br>The data from textfield is " +  
    request.getParameter("textfield")) ;  
out.println("<br>The data from textarea is " +  
    request.getParameter("textarea")) ;  
String[] course = request.getParameterValues("course");  
out.println("<br>The data from checkbox are<br>");  
for(int i=0; i< course.length; i++)  
    out.println(course[i] + "<br>");  
out.println("<br>The data from radiobutton is " +  
    request.getParameter("degree");  
out.println("<br>The data from single-list is " +  
    request.getParameter("department");  
out.println("<br>The data from multi-list is as follows:<br>");  
String[] sport = request.getParameterValues("sport") ;  
for(int i=0; i<sport.length; i++)  
    out.println(sport[i] + "<br>");  
%>
```

```
<%@ page import="java.util.*" %>
:
Current time is <br>
<%
    Date date = new Date() ;
    out.println((date.getYear()+1900) + "-" +
        (date.getMonth()+1) + "-" +
        date.getDate() + "(" + date.getDay() + ") " +
        date.getHours() + ":" +
        date.getMinutes() + ":" +
        date.getSeconds() + "<br>");
%>
Server Information<br>
IP address is <%=request.getLocalAddr()%> <br>
Server is <%=request.getServerName()%> <br>
Browser is <%=this.getServletContext().getServerInfo()%> <br>
Current document is <%=request.getPathInfo()%> <br>
Web page root is <%=this.getServletContext().getContext()%> <br>
Server software is <%=this.getServletContext().getServerInfo()%><br>
```

將表單資料傳至伺服器端

文字欄位

文字框測試

文字區域

第一段

第二段

核取方塊

計概 微積分 英文 國文

選項按鈕

學士 碩士 博士

選單

中資系

清單

棒球
壘球
籃球
排球
羽球

送出

重設

Client Site

主從式架構 資料傳送

接收資料 伺服器資訊

Server Site

Current time is 02:13
IP address is 127.0.0.1
Server is localhost
Browser is Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-TW; rv:1.8.1.6) Gecko/20070725 Firefox/2.0.0.6
Current document is /php/test/FormTest.php
Web page root is C:/Apache2.2/htdocs
Server software is Apache/2.2.2 (Win32) PHP/5.1.6

The data from textfield is 文字框測試
The data from textarea is 第一段第二段

The data from checkbox are

Computer
Calculus
English

The data from radiobutton is BS

The data from single-list is 中藥資源

The data from multi-list is Array as follows:

壘球
籃球
排球

傳輸參數類型

- 傳輸方式(method)
 - GET: 按提交鈕或Enter傳送時會在網址輸入列顯示傳送的參數值
 - POST: 按提交鈕或Enter傳送時會隱藏參數值
- 參數類型(type)
 - 一般文字格(text)：未輸入時為空字串
 - 多行文字區塊(textarea)：未輸入時為空字串
 - 單選鈕(radio)：未點選時為空值
 - 核取方塊(check box)：未點選時為空值
 - 可複選核取方塊未選擇時為空值
 - 下拉式選單(drop-down list)：未選擇時以第一個選項值為輸入值
 - 可複選下拉式選單未選擇時為空值
- 按鈕類型(button)
 - 提交鈕(submit)：送出資料
 - 重設鈕(reset)：回覆至預設狀態重新輸入資料
 - 一般按鈕(button)：配合JavaScript使用

變數與資料型態

- 命令句
 - 在JSP中，每一個命令句必須以分號；作結束。
- 註解列(Comment Line)
 - 註解列是在說明程式內容，不會予以執行，JSP的註解方式與C/C++/Java相同
 - 單行註解：`//` 單行說明文字
 - 多行註解：`/*` 多行註解區段 `*/`
- 變數(Variable)
 - 變數是用來儲存程式執行時的暫存資料。
 - JSP的大小寫是有差別的(Case Sensitive)，故使用變數時，須注意大小寫是否一致。
 - JSP是屬於strongly typed的程式語言，對資料型態的規範嚴格，未經宣告的變數，無法使用。
- 常數(Constant)
 - 常數表不會變動的值，利用final定義：`final` 資料型態 常數名 = 常數值
 - 例如

```
final int price = 2500 ;
final String tirebrand = "GoodYear";
```


變數與資料型態

- 資料型態(Data Type)

- 整數(int)：不帶小數點的數字，如0, 10, -5
- 浮點數(float, double)：有帶小數點的數字，如3.5234, -23.43
- 布林變數(boolean)：邏輯判斷值，true或false
- 字元(char)：字母資料，以單引號(')標示，如'5','C'
- 字串(String)：文字資料，以雙引號(")標示，如"中國醫大", "1.54", "Taiwan"

- 資料型別轉換(Data Converse)

- (資料型別)變數名稱 或 轉換函式(資料或變數)
- 例如，data = 67.52 → 為實數(浮點數)，則
(int)data → 但會喪失精準度
String.valueOf(data) → "67.52"
Double.parseDouble("67.52") → 67.52
Integer.parseInt("50") → 50
- 相同資料型態的變數方可進行運算，否則要先經過型別轉換。

變數生命週期

- 全域變數與區域變數生命週期
 - 所有變數都是獨立的，在其生命週期內不可重複使用
 - 在最外層區段定義之變數可適用於所有區段，稱之為全域變數
 - 位於{ }區段內的變數，只能在此區段內之各區段使用，為區域變數
 - 如下例錯誤示範中，假設變數money在此之前尚未被定義，則只能在第一個{ }內有效，在第二個{ }內則無法使用；同理j在內層的{ }可以輸出，但j在外層的{ }則失效。

```
If (true)
{
    int money = 100 ;
    out.println(money) ;
}
else
{
    out.println(money);
}
```

錯誤

```
for (int i=0; i<10; i++)
{
    out.println(j) ;
    for (int j=0; j<10; j++)
    {
        out.println(i * j);
    }
}
```

基本輸出

- 基本輸出函式 `out.println`
 - `out.println` 可將變數及資料轉換為字串輸出，不同變數資料間用加號(+)加以連接
 - 語法
`out.println(輸出資料一 + 輸出資料二);`
- 保留字與跳脫字元
 - 定義變數時，不可使用 `jsp` 預設的保留字
 - 如欲輸出這些保留字，則須在前面加上跳脫字元 `\`，例如
`money=100;`
`out.println("今天賺了\"新台幣\"$"+money);`
輸出則為：今天賺了"新台幣" \$100

| 跳脫字元 | 代表意義 | 跳脫字元 | 代表意義 |
|-------------------------|--------------|--------------------------|--------------|
| <code> </code> | 在網頁文件中換行 | <code>\r</code> | 歸位字元 |
| <code>\"</code> | 顯示雙引號符號" | <code>\t</code> | 進格字元 |
| <code>\\</code> | 顯示倒斜線符號\ | <code>\000 ~ \377</code> | 以16進位表示某一個字元 |
| <code>\n</code> | 換行字元 | <code>\x00 ~ \xFF</code> | 以8進位表示某一個字元 |

運算元(Operator)

- 數學運算元

| 運算元 | 意義 | 範例 | 運算元 | 意義 | 範例 |
|-----|----|---------|-----|----|-------------------|
| + | 加號 | $a + b$ | % | 餘數 | $10 \% 7$ (得餘數為3) |
| - | 減號 | $a - b$ | ++ | 加一 | 將變數加一 |
| * | 乘號 | $a * b$ | -- | 減一 | 將變數減一 |
| / | 除號 | a / b | | | |

- 邏輯運算元

| 運算元 | 代表意義 | 用法 | 解說 |
|-----|-------|--------------|--------------------------|
| and | AND運算 | $a \& b$ | 當二個變數皆為真(True)時成立 |
| or | OR運算 | $a b$ | 只要有一變數為真則成立 |
| xor | XOR運算 | $a \wedge b$ | 只一變數為真且另一變數為假(False)時才成立 |
| ! | NOT運算 | $!a$ | 若變數為真，NOT運算後為假，反之亦然 |
| && | AND | $a \&\& b$ | 當二個變數皆為真(True)時成立 |
| | NOT | $a b$ | 只要有一變數為真則成立 |

● 比較運算元

| 運算 | 意義 | 解說 |
|---------------------------|------------|---|
| <code>a == b</code> | 相等 | <code>a</code> 、 <code>b</code> 相等時成立 |
| <code>a === b</code> | 相等且同型態 | <code>a</code> 、 <code>b</code> 相等且型態相等時成立 |
| <code>a != b</code> | 不相等 | <code>a</code> 、 <code>b</code> 不相等時成立 |
| <code>a <> b</code> | 不相等 | <code>a</code> 、 <code>b</code> 不相等時成立 |
| <code>a !== b</code> | 不相等且型態也不相等 | <code>a</code> 、 <code>b</code> 不相等且型態也不相等時成立 |
| <code>a < b</code> | 小於 | <code>a</code> 小於 <code>b</code> 時成立 |
| <code>a > b</code> | 大於 | <code>a</code> 大於 <code>b</code> 時成立 |
| <code>a <= b</code> | 小於或等於 | <code>a</code> 小於或等於 <code>b</code> 時成立 |
| <code>a >= b</code> | 大於或等於 | <code>a</code> 大於或等於 <code>b</code> 時成立 |

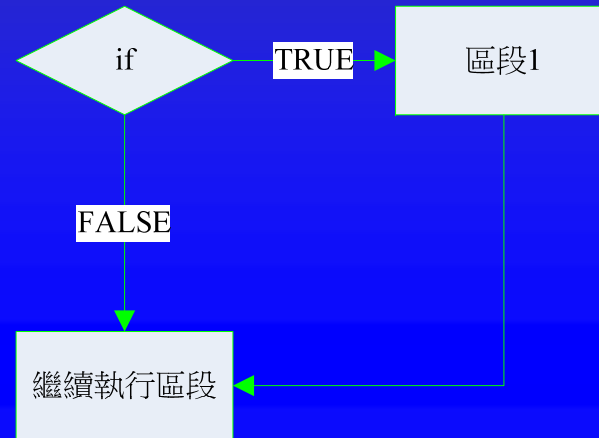
● 運算元優先順序

| 優先順序：由左而右由上而下 | | | |
|---|--|---------------------------------|---|
| <code>++</code> 、 <code>--</code> 、 <code>~</code> 、轉型運算子 | <code>*</code> 、 <code>/</code> 、 <code>%</code> | <code>+</code> 、 <code>-</code> | <code><</code> 、 <code><=</code> 、 <code>></code> 、 <code>>=</code> |
| <code>==</code> 、 <code>!=</code> | <code>&</code> 、 <code>&&</code> | <code>^</code> | <code> </code> 、 <code> </code> |

控制流程 - 邏輯判斷

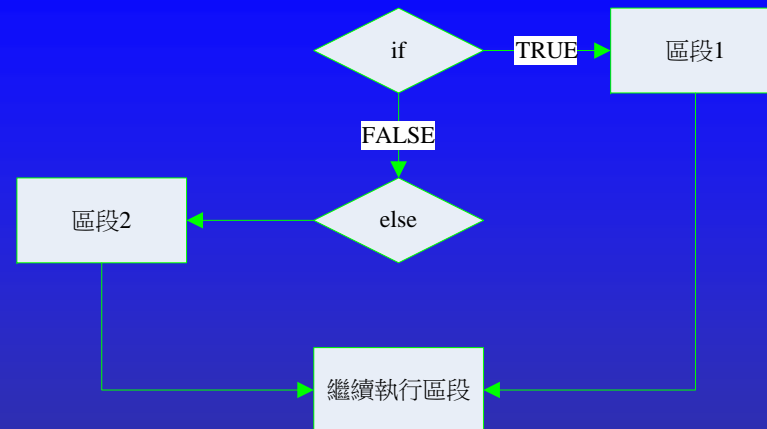
- if

- if (比較式)
{
 比較式成立時
 執行此區段;
}



- if ... else

- if (比較式){
 執行區段1;
} else {
 執行區段2;
}



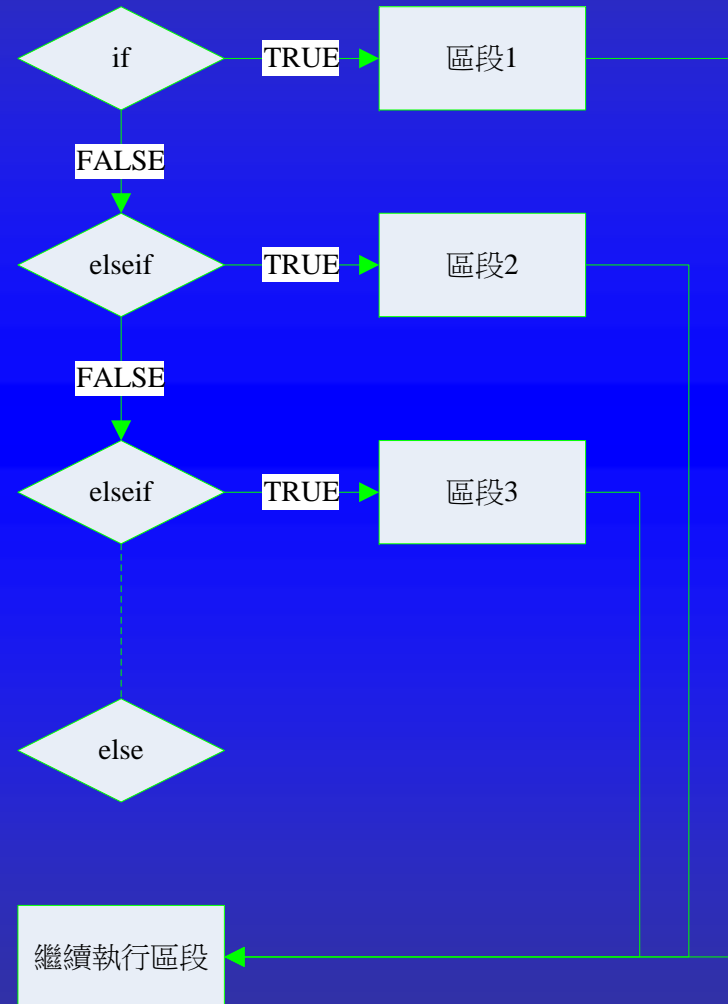
控制流程 - 邏輯判斷

- **if ... else if ... else**

- if (比較式1){
 執行區段1
}elseif (比較式2){
 執行區段2
}else{
 執行區段
}

- **switch**

- switch(變數)
 {
 case 比較值:
 執行區段;
 break;
 :
 default:
 執行區段;
 break;
 }



控制流程 - 邏輯判斷範例

```
if(fruit.equals("apple")){  
    myfruit = "蘋果" ;  
}
```

```
if(fruit == null){  
    myfruit = "未輸入" ;  
}
```

```
switch(fruitID){  
    case 'A':  
        myfruit = "蘋果";  
        break;  
    case 'O':  
        myfruit = "柳橙";  
        break;  
    default:  
        myfruit = "西瓜";  
}
```

```
if(fruit.equals("apple")){  
    myfruit = "蘋果" ;  
}else{  
    myfruit="西瓜" ;  
}
```

```
if(fruit != null){  
    myfruit = "有輸入資料" ;  
}
```

```
if(fruit.equals("apple")){  
    myfruit = "蘋果" ;  
}else if(fruit.equals("orange")){  
    myfruit == "柳橙" ;  
}else{  
    myfruit="西瓜" ;  
}
```


控制流程 - 迴圈結構

- **for**迴圈

- for(記數起始條件; 記數終止條件; 記數累進) {
 執行區段;
}

- **while**迴圈

- while(判斷式){
 執行區段;
}

```
out.println("<table>");
for(int row=1; row<=9; row++) {
    out.println("<tr>");
    for(int col=1; col<=9; col++) {
        out.println("<td>" + row + "x" +
col + "=" + row*col + "</td>");
    }
    out.println("</tr>");
}
out.println("</table>");
```

```
out.println("<table>");
int row=1;
while(row<=9){
    out.println("<tr>");
    int col=1;
    while(col<=9){
        out.println("<td>" + col +
"x" + row + "=" + col*row +
"</td>");
        col++;
    }
    out.println("</tr>");
    row++;
}
out.println("</table>");
```

控制流程 – 迴圈結構

- **do**迴圈

- do{
 執行區段;
} while(判斷式);

- **Foreach**迴圈(配合陣列)

- for(宣告變數:變數所屬陣列)
 執行區段;
 (變數為陣列中元素)
}

- **break** – 離開迴圈

- **continue** – 繼續迴圈

```
String[] sport = {"baseball",  
    "basketball", "football", "hockey"};  
for(int i=0; i<sport.length; i++)  
    out.println(sport[i] + "<br>");
```

```
out.println("<table>");  
int row=1;  
do{  
    out.println( "<tr>");  
    int col=1;  
    do{  
        out.println("<td>" + col + "x" + row +  
            "=" + col*row + "</td>");  
        col++;  
    }while(col<=9);  
    out.println("</tr>");  
    row++;  
}while(row<=9);  
out.println( "</table>");
```

陣列資料型態

- 使用時機
 - 用於記錄多筆相同資料型態的變數
- 一維陣列宣告語法
 - 資料型態[] 變數名 = {值一,值二,值三,...};
 - 資料型態[] 變數名 = new 資料型態[陣列大小];
- 二維陣列宣告語法
 - 資料型態[][] 變數名 = {{值一之一,值一之二,值一之三,...},{值二之一,值二之二,值二之三,...},...};
 - 資料型態[][] 變數名 = new 資料型態[陣列大小][陣列大小];

陣列資料型態

- 一維陣列宣告範例

- `String[] fruit = {"apple", "orange", "cherry"};`
 - 直接賦予初始值，陣列大小不能再修改，初始值不能是變數
- `String[] fruit = new String[3];`
`fruit[0] = "apple"; fruit[1] = "orange"; fruit[2] = "cherry";`
- `String[] fruit = new String[]{"apple", "orange", "cherry"};`
 - 先宣告新物件再賦予初始值，陣列大小可調整，初始值可以是變數

- 二維陣列宣告範例

- `String[][] car = {"Accord", "Civic", "CRV"}, {"Camry", "Vios", "RAV"};`
 - 亦可先宣告物件再賦予初始值，陣列大小可調整，初始值可以是變數
- `String[][] car = new String[2][];`
`String[] honda = {"Accord", "Civic", "CRV"};`
`String[] toyota = {"Camry", "Vios", "RAV"};`
`car[0] = honda; car[1] = toyota;`
- `String[][] car = new String[2][3];`
`car[0][0] = "Accord"; car[0][1] = "Civic"; car[0][2] = "CRV";`
`car[1][0] = "Camry"; car[1][1] = "Vios"; car[1][2] = "RAV";`

表單資料空值處理

- 客戶端網頁表單資料未填寫時的預設值
 - 在輸入格的value屬性設預設值
 - 文字輸入框(text或textarea)的value屬性預設為空字串""
 - 核取鈕(radio)、核取方塊(checkbox)的value屬性若未特別設定強迫勾選(即在選項處加上checked屬性)則預設為空值null
 - 下拉式選單(select)如未特別設定(即在選項加上屬性selected)，其value屬性單選時預設為第一個選項，複選時為空值null
- 伺服器端變數取得表單資料為空值時，即未初始化
 - 不可進行字串轉碼，應配合if..else控制賦予初始值
 - 輸入資料有可能是中文時(如文字框)需轉碼，例如

```
str = new String(str.getBytes("8859_1"),"UTF-8");
```

客戶端表單

```
<form action="process.jsp" method="POST">
  <p>
    文字方塊:<input type="text" name="TEXT" value="" /><br>
    留言區:<br><textarea name="TEXTAREA" rows="4" cols="20"></textarea>
    <br>
    核取鈕:
    <input type="radio" name="RADIO" value="A" />Radio A
    <input type="radio" name="RADIO" value="B" />Radio B
    <input type="radio" name="RADIO" value="C" />Radio C
    <br>
    核取方塊:
    <input type="checkbox" name="CHECKBOX" value="1" />Check 1
    <input type="checkbox" name="CHECKBOX" value="2" />Check 2
    <input type="checkbox" name="CHECKBOX" value="3" />Check 3
    <br>
    單選單:
    <select name="S_SELECT">
      <option value="TAIWAN">台灣</option>
      <option value="JAPAN">日本</option>
      <option value="KOREA">韓國</option>
    </select>
    <br>
    複選單:<br>
    <select name="M_SELECT" multiple="multiple">
      <option value="蘋果">APPLE</option>
      <option value="香蕉">BANANA</option>
      <option value="桃子">PEACH</option>
    </select>
  </p>
  // continue →
```

文字方塊:

留言區:

核取鈕: Radio A Radio B Radio C

核取方塊: Check 1 Check 2 Check 3

單選單: 台灣

複選單:
APPLE
BANANA
PEACH

Submit Reset

```
<%
  Date date = new Date();
%>
<input type="hidden" name="HIDDEN"
value="<%=date%>" />
<p>
<input type="submit" value="Submit" />
<input type="reset" value="Reset" />
</p>
</form>
```

• 接收客戶端表單資料

```
<%
    String text = request.getParameter("TEXT");
//此設計一定不為空值, 可進行轉碼
    text = new String(text.getBytes("8859_1"), "UTF-8");
    String textarea = request.getParameter("TEXTAREA");
//此設計一定不為空值, 可進行轉碼
    textarea = new String(textarea.getBytes("8859_1"), "UTF-8");
//前端頁面設計有可能都未勾選, 如有勾選時傳輸資料為英文, 無需轉碼
    String radio = request.getParameter("RADIO");
//前端頁面設計有可能都未勾選, 如有勾選時傳輸資料為數字字串, 無需轉碼
    String[] checkbox = request.getParameterValues("CHECKBOX");
//前端頁面設計第一個選項為預設值, 且為英文或數字, 無須轉碼
    String s_select = request.getParameter("S_SELECT");
    String[] m_select = request.getParameterValues("M_SELECT");
//前端頁面設計有可能都未勾選, 如有勾選時則為中文字, 必須在有勾選的情況下轉碼
    if (m_select != null) {
        for (int i = 0; i < m_select.length; i++) { //將陣列內元素逐一轉碼
            m_select[i] = new String(m_select[i].getBytes("8859_1"), "UTF-8");
        }
    }
    String date = request.getParameter("HIDDEN");//隱藏傳輸資料, 設定為時間自串
%>
```

伺服器端控制(cont.)

- 控制空值資料並輸出

```
<p>連線時間: <%=date%></p>
<p>文字方塊資料:
<%
  if (text.equals("")) {
    out.println("輸入資料為空字串");
  } else {
    out.println(text);
  }
%>
</p>
<p>留言板資料:
<%
  if (textarea.equals("")) {
    out.println("輸入資料為空字串");
  } else {
    out.println(textarea);
  }
%>
</p>
```

```
<p>核取鈕資料:
<%
  if (radio != null) {
    out.println(radio);
  } else {
    out.println(radio + "(未選取資料)");
  }
%>
</p>
<p>核取方塊資料:
<%
  if (checkbox != null) {
    for (String check : checkbox) {
      out.println(check + ", ");
    }
  } else {
    out.println("null(未選取資料)");
  }
%>
</p>
```


伺服器端控制(cont.)

- 控制選單空值資料並輸出
- 利用表單按鈕指定跳頁

```
<p>單選資料:  
<%  
    if (s_select != null) {  
        out.println(s_select);  
    } else {  
        out.println(s_select + "(未選取資料)");  
    }  
<%>  
</p>  
<p>多選資料:  
<%  
    if (m_select != null) {  
        for (String select : m_select) {  
            out.println(select + ", ");  
        }  
    } else {  
        out.println("null(未選取資料)");  
    }  
<%>  
</p>
```

連線時間: Sat Sep 05 11:08:00 CST 2015

文字方塊資料: (輸入資料為空字串)

留言板資料: (輸入資料為空字串)

核取鈕資料: null(未選取資料)

核取方塊資料: null(未選取資料)

單選資料: TAIWAN

多選資料: null(未選取資料)

(回到前頁並保留填寫的資料)

(回到首頁重新刷新頁面)

```
<p>  
<input type="button" value="回前一頁"  
onclick="history.back()" />  
(回到前頁並保留填寫的資料)  
</p>  
<p>  
<input type="button" value="回到首頁"  
onclick="javascript:location.href = 'index.jsp'" />  
(回到首頁重新刷新頁面)  
</p>
```

檔案資料處理

- 匯入套件
 - `<%@page import="java.io.*, java.util.*"%>`
- 開啟檔案
 - `File f = new File("檔案位置");`
 - 檔案位置必須為實際路徑+檔名，實際路徑可用`request.getRealPath("/")`獲得
 - `f.exists()` → 檔案是否存在
- 讀取檔案
 - `Scanner sc = new Scanner(f);`
 - 讀取單行：`sc.nextLine();`
 - 讀取字串：`sc.next();`
 - 是否還有下一行：`sc.hasNext();` // 一般與while迴圈合用
- 寫入檔案
 - `FileWriter fw = new FileWriter(f);` // 建立新檔或覆蓋原有檔案
 - `FileWriter fw = new FileWriter(f, true);` // 將新資料附加至原有檔案
 - `fw.write(.....);` // 與`out.println()`相同，但將資料寫入檔案中
 - `FileWriter`涉及資料寫入動作，使用時必須在`try { ... } catch { ... }`區塊中執行
- 關閉檔案
 - `fw.close();`

檔案管理

- 瀏覽檔案(File物件變數為f)
 - 檔案是否存在：f.exists()
 - 檢查檔案大小：f.length()
 - 刪除檔案：f.delete()
 - 其他：f.isFile(), isDirectory(), isHidden()..., etc.
- 上傳檔案
 - 表單資料格式(in html form)

```
<form ... enctype="multipart/form-data">  
<input type="file" name="檔案上傳變數">
```
 - 擷取檔案資訊(in JSP)
需使用其他Java函式庫套件如cos.jar進行檔案資料串流的擷取

檔案讀取與寫入

```
<%
    String path =
request.getRealPath("/") ;
    File f = new
File(path+"test.txt");
    String bak = "test.bak";
    Scanner sc = new Scanner(f) ;
try{
    FileWriter fw = new
FileWriter(path+bak);
    while(sc.hasNext()){
        String str = sc.next();
        fw.write(str);
    }
    fw.close();
} catch() {}
%>
```

瀏覽檔案

```
String path = request.getRealPath("/") ;
String dir = "test" ; // the folder under current
path
File f = new File(path + dir) ;
String[] files = f.list() ;
for(int i=0; i<files.length; i++) {
    String str = "" ;
    File fname = new File(files[i]) ;
    if(fname.canRead())
        str = str + "r" ;
    if(fname.canWrite())
        str = str + "w" ;
    if(fname.isDirectory())
        str = str + "d" ;
    out.println(fname.getName() + "-" + str + "-
<br>");
    out.println(fname.getLastModified() +
"<br>");
    out.println(fname.length() + "<br>");
}
```

連線資訊處理

- 資料傳遞連線容器

- 線上資料傳遞時，先將資料暫存至指定的容器，在委派到指定頁面，常用的有request和session
- 資料可為陣列、集合、字串資料等物件
- 避免傳送單一數值，可先存至陣列或集合
- 把資料委派至指定的JSP頁面，使用request，傳遞完成後資料便從容器中移除，例如，`request.setAttribute(...)`
- 希望資料能在連線期間讓各JSP頁面擷取，使用session，在連線未結束前資料一直存在容器中，例如，`session.setAttribute(...)`
- 存到容器：`request.setAttribute(“容器名稱”,物件名稱);`
- 取出資料：`(物件資料型態)request.getAttribute(“容器名稱”);`

- 網頁重導(Redirect)與委派(Dispatch)

- `response.sendRedirect(“網頁或網址”)`
- `RequestDispatcher dispatch = request.getRequestDispatcher(“網頁名稱”);`
`dispatch.forward(request,response);`
- 重導或委派前後不可印出資料

```
<form action="LoginCheck.jsp" method="post">
User:<input type="text" name="username">
Pass:<input type="password" name="password">
<input type="submit" value="送出">
<input type="reset" value="重設"></form>
```

資料
輸入頁

```
<%
String username =
request.getParameter("username");
String password =
request.getParameter("password");
if (username.equals("") ||
password.equals("")){
    response.sendRedirect("error.html");
}else{
    String[] userpass = {username,
password};
request.setAttribute("user",userpass) ;
RequestDispatcher dispatch =
request.getRequestDispatcher("result.js
p");
dispatch.forward(request,response) ;
}
%>
```

驗證資料 → 頁面不顯示

```
<html>
<body>
<h2>
Please enter username
and password!
</h2>
</body>
</html>
```

錯誤頁面

驗證失敗(重導)

驗證成功(委派)

```
<%
String[] userpass =
(String[])request.getAttribute("us
er");
for(int i=0; i<userpass.length;
i++)
    out.println(userpass[i] +
"<br>");
%>
```

呈現頁面

子程式與函式

- 定義函式(Function)
 - 定義語法
函式形態 函數名(){...}
 - 呼叫函數
函數名();
- 參數(Parameter)傳遞
 - 定義語法
函式形態 函數名(參數1, 參數2, ...){...}
 - 呼叫函數
函數名(變數1, 變數2, ...);
- 具傳回值函數
 - 定義語法
函式形態 函數名(參數, ...) {
...
return 傳回值;
}
 - 呼叫函數
變數 = 函數名(參數, ...);

子程式與函式範例

```
<%!  
double Area(r) {  
    double a = Math.PI*r*r ;  
    return a ;  
}  
double Circum(r) {  
    double c = 2*Math.PI*r ;  
    return c ;  
}  
String error() {  
    out.println("Radius is missing") ;  
}  
%>
```

子程式區塊可獨立出現在
JSP 頁面程式碼的任何位置

```
<%  
double radius = 10.5;  
if (radius > 0){  
    double area = Area(radius) ;  
    double circum = Circum(radius) ;  
    out.println("Area: " + area + "<br>") ;  
    out.println("Circumference: " + circum + "<br>") ;  
}else{  
    out.println(error() + "<br>");  
}  
%>
```