

資料輸入表單與 **request** 物件

request 物件

- request 物件代表連線時，由客戶端送至伺服端的資料。
- request 物件將實作 `javax.servlet.http.HttpServletRequest` 介面。

表單的建立與資料傳遞

- 資料的傳遞過程

- 整個運作的過程為使用者透過網頁中的表單，將資料傳送給 Web 伺服器，並呼叫 **Response.jsp** 檔。
- 此時， **request** 物件（ **JSP** 物件）取得使用者由表單傳送過來的資料。

表單的建立與資料傳遞

- 表單的建立語法

- 顯示於客戶端，供使用者輸入資料的表單，必須使用 **HTML 語法** 中的 **<FORM>** 標籤建立，並完成下表三種屬性的設定。

屬性	說明
Action	此表單傳出資料後，執行回應動作的 JSP 檔。
Name	表單的名稱。
Method	傳遞資料的方式，有 Post 與 Get 兩種。

表單的建立與資料傳遞

- 表單的建立語法

- 表單的建立語法如下：

**<FORM action=JSP 名稱路徑 method=
{post | get} name= 表單名稱 >**

表單中的控制項

</FORM>

表單的建立與資料傳遞

- 表單的建立語法

- 在表單中控制項的類型，主要有以下六種：
 - ✓ 指令按鈕
 - ✓ 文字方塊
 - ✓ 文字區控制項
 - ✓ 下拉選單
 - ✓ 多選鈕
 - ✓ 核取方塊

表單的建立與資料傳遞

- 取得客戶端傳遞的資料

- 在 JSP 檔中，取得客戶端所傳來資料的動作，將透過 request 物件的 `getParameter` 方法完成，其語法如下：

request.getParameter(控制項名稱)

表單的建立與資料傳遞

- 資料傳遞的 Get 方式

- Post 請求的架構圖：



在 **Response.jsp**
中，取得表單傳回
資訊的方法均相同



- Get 則會將傳遞的訊息顯示在欲瀏覽網頁網址後，
架構圖

表單的建立與資料傳遞

- 資料的轉換

- 當利用 **request** 物件，取得從表單中傳遞過來的資料時，所取得資料之型別均為 **String** 。
- **JSP** 網頁需要處理數值運算時，必須將字串轉換成數值，使用 **JAVA** 語法中的方法即可。

型別類別 **.parseXXX(String 字串)**

表單的建立與資料傳遞

- 資料的轉換 (將字串轉換為數值)

- 將字串 **true** 或 **false** , 轉換為 **boolean** 值時, 則必須利用 **Boolean** 類別的 **valueOf** 方法, 該方法為類別方法, 其語法如下:

Boolean.valueOf(String 字串)

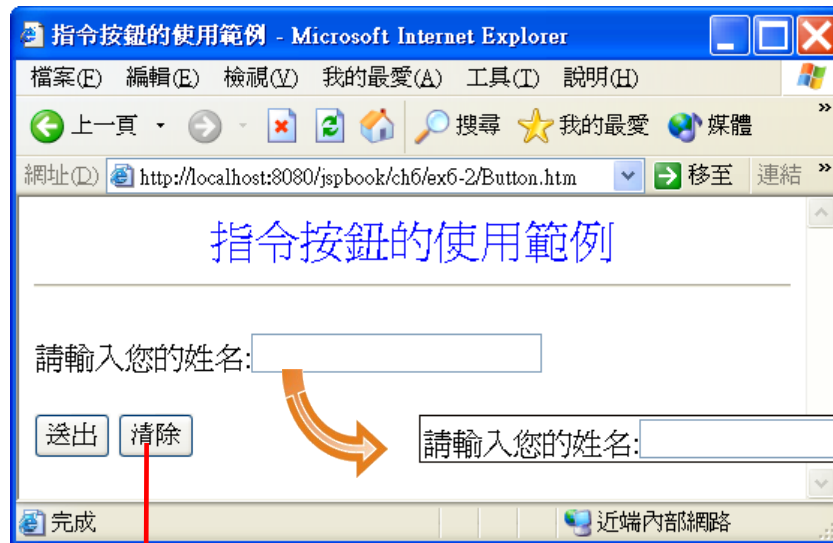
- 下面是語法各部份的說明:
 - ✓ 字串

欲轉換為 **boolean** 值的字串, 若該字串為 " **true** ", 則 **valueOf** 方法將傳回 **true** 。若不是 " **true** ", 則傳回 **false**

控制項的建立與訊息傳遞

- 指令按鈕

- 在表單中，須要使用的指令按鈕有 **Submit 按鈕（送出）** 與 **Reset 按鈕（重設）** 兩種，後者用於清除表單中已輸入的資料，供使用者再次輸入。



① 按下

② 文字方塊中的資料被清空

控制項的建立與訊息傳遞

- 指令按鈕

- 建立指令按鈕時，需使用 HTML 語法的 **<Input>** 標籤，且需要設定下表的幾個屬性：

屬性	說明
Type	指令按鈕的類型，設定值為 Reset（重設）、Submit（送出）。
Name	指令按鈕的名稱。
Value	此值為顯示在按鈕上的字串。

其語法如下：

<Input Type = {submit | reset}

Name= 按鈕名稱 Value= 顯示值 >

控制項的建立與訊息傳遞

- 文字方塊與密碼輸入欄

- 文字方塊與密碼欄的 HTML 語法如下：

<Inupt **Type = {text | password}** **Name=** 文字方塊名稱 **Value=** 傳出值 **>**

- 各屬性的意義如下表：

屬性	說明
Type	設定控制項的類型，建立文字方塊時，此屬性設為 text 。建立密碼輸入欄時，設定為 password 。
Name	控制項的名稱。
Value	傳出值，若設定此值時，此值將顯示在文字方塊中，達到預設輸入值的效果。當按下表單中 Reset 類型的指令按鈕，進行重設時，將使文字方塊恢復至此值。但密碼輸入欄通常不設定。

控制項的建立與訊息傳遞

- 文字區控制項

- 文字區控制項的 HTML 語法如下：

<TEXTAREA rows= 列數 cols= 行數 name
= 控制項名稱 >

預設輸入值

</TEXTAREA>

控制項的建立與訊息傳遞

- 文字區控制項

- 下表為語法中各部份的說明：

名稱	說明
Rows	文字區的高度設定，其單位為文字的列數。
Cols	文字區的寬度設定，其單位為文字的列數。
Name	文字區控制項的名稱。
預設輸入值	在兩 <TEXTAREA> 標籤所輸入的值，將顯示在文字區中，可達到類似預設值的效果。

控制項的建立與訊息傳遞

- 下拉清單方塊

- 建立下拉清單方塊控制項的 HTML 語法如下：

```
<SELECT name= 控制項名稱 size=1>  
  <OPTION value= 傳出值 1 selected> 選項 1  
                                     </OPTION>  
  <OPTION value= 傳出值 2> 選項 2</OPTION>  
  .....  
  <OPTION value= 傳出值 N> 選項 N</OPTION>  
</SELECT>
```


控制項的建立與訊息傳遞

- 下拉清單方塊

- 語法中各部份的說明，請參考下表：

名稱	說明
Name	控制項的名稱。
Size	控制項的大小，建立下拉清單方塊時，必須設為 1。
Selected	設定預設被選取的選項。
Value	該選項被選取後的傳出值，若未設定則傳出值，即為顯示在清單中的選項值。
選項 1...N	顯示在下拉清單方塊之選單中的選項名稱。

控制項的建立與訊息傳遞

- 清單方塊

- 清單方塊與下拉清單方塊一樣，是由 **<SELECT>** 標籤與 **<OPTION>** 標籤所組成。
- 不同的地方：
設定 **<SELECT>** 標籤的 **size** 屬性，其值不可設為 **1**，如此便可建立出清單方塊。

控制項的建立與訊息傳遞

- 清單方塊

- 以下為建立清單方塊的 HTML 語法:

**<SELECT name= 控制項名稱 size= 控制項大小
Multiple>**

**<OPTION value= 傳出值 1 selected> 選項 1
</OPTION>**

<OPTION value= 傳出值 2> 選項 2</OPTION>

.....

<OPTION value= 傳出值 N> 選項 N</OPTION>

</SELECT>

控制項的建立與訊息傳遞

- 清單方塊

- 語法中各部份的說明，請參考下表：

名稱	說明
Name	控制項的名稱。
size	下拉清單的大小，必須設為大於 1 的值。
Multiple	設定清單方塊中的選項是否可被多重選取。
selected	預設被選取的選項。
value	該選項被選取後的傳出值，若未設定則傳出值即為顯示在清單中的選項值。
選項 1...N	顯示在清單方塊中的選項名稱。

控制項的建立與訊息傳遞

- 清單方塊

- 在回應的 JSP 網頁中，取得單選型清單方塊所傳出的值。若傳出的是字串，則必須利用 **if... else if** 敘述，配合 **String** 物件的 **equals** 方法，判斷傳出值。
- 若傳出值為字元或者數值時，則可**利用 switch 判斷敘述執行比對**。當為數值則必須在取得清單方塊值時，將字串轉換為數值

控制項的建立與訊息傳遞

- 清單方塊

- 多選型的清單方塊 -

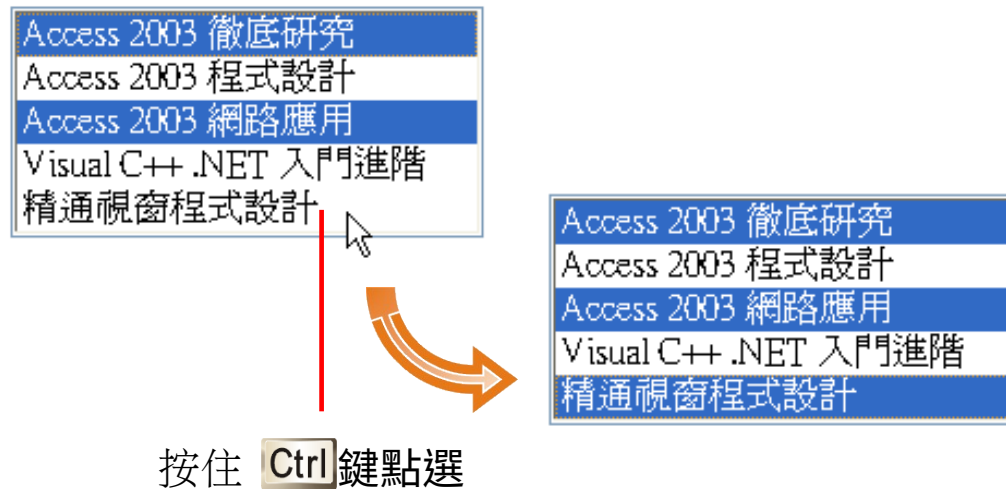
下面 HTML 敘述，將建立一個可多重選取選項的清單方塊。

```
<SELECT name=lbxBook size=5 Multiple>  
  <OPTION value = "A9073" selected>Access 2000 徹底研究  
</OPTION>  
  <OPTION value = "A9193" >Access 2000 程式設計 </OPTION>  
  <OPTION value = "A0053" selected >Access 2000 網路應用  
</OPTION>  
  <OPTION value = "P9067" >Visual C++ 入門進階 </OPTION>  
  <OPTION value = "P9127" >精通視窗程式設計 </OPTION>  
</SELECT>
```

控制項的建立與訊息傳遞

- 清單方塊

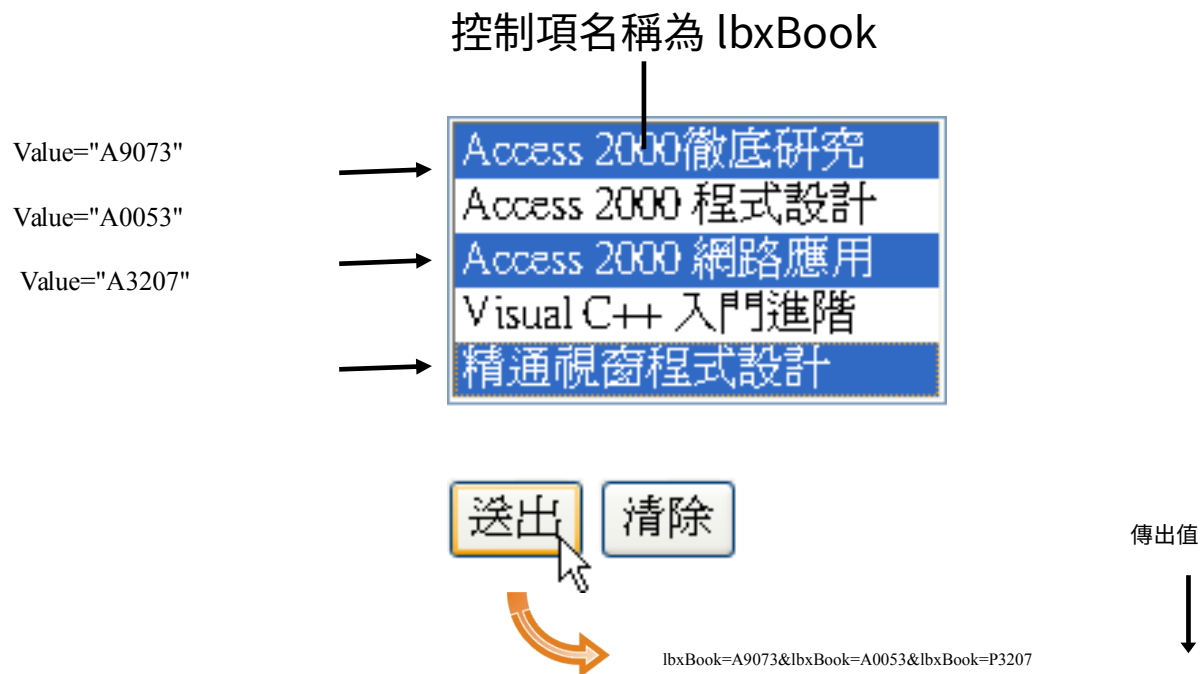
- 欲多重選取選項時，只要**按住鍵盤的 Ctrl 按鍵**，然後用滑鼠點選欲選取的選項即可：



控制項的建立與訊息傳遞

- 清單方塊

- 多選型清單方塊的傳出值，有點像把所有被選取選項，都當成一個單獨控制項，並**連同控制項名稱，串聯成一字串**。



控制項的建立與訊息傳遞

- 清單方塊

- 若要取得清單方塊的傳出值時，必須使用 `request` 物件的 `getParameterValues` 方法。
語法如下：

字串陣列變數 = `request.getParameterValues`
(" 複選型清單方塊名稱 ");

控制項的建立與訊息傳遞

- 清單方塊

- 轉成字串陣列後，通常都會利用一個 **for 迴圈**，將陣列的字串，一個一個取出來使用。
- 最大索引值可利用 `java.lang.reflect` 套件中，`Array` 類別的 **`getLength`** 方法取得。
變數 `= java.lang.reflect.Array.getLength (" 陣列名稱 ");`

控制項的建立與訊息傳遞

- 選擇鈕

- 在網頁中，選擇鈕通常用於供使用者從多個選項中，選取一個選項。建立選擇鈕的 HTML 語法如下：

<Input Type = radio Name= 選擇鈕名稱 Value = 傳出值 checked>

— 各屬性的意義請參考下表：

屬性	說明
Type	設定控制項的類型，建立選擇鈕時，此屬性設為 radio。
Name	控制項的名稱。若要建立一組選擇鈕時，只要將各選擇鈕設為同一名稱即可。
Value	此選擇鈕被選取時的傳出值。
checked	此選擇鈕是否預設為選取。

控制項的建立與訊息傳遞

- 核取方塊

- 在網頁中，核取方塊用於供使用者選取某選項用。建立核取方塊的 HTML 語法如下：

<Input Type = checkbox Name= 核取方塊名稱 checked>

各屬性的意義，說明如下表：

屬性	說明
Type	設定控制項的類型，建立核取方塊時，此屬性設為 check。
Name	控制項的名稱。
Checked	使用此屬性時，將設定此核取方塊為 預設選取 。

控制項的建立與訊息傳遞

- 核取方塊

- 當核取方塊被點選後，將會傳出 **on** 字串；若未被點選，傳出值為 **null**。
- 要比對核取方塊的傳出值是否為 **null**。若不等於 **null**，即代表該核取方塊被選取。

其語法如下：

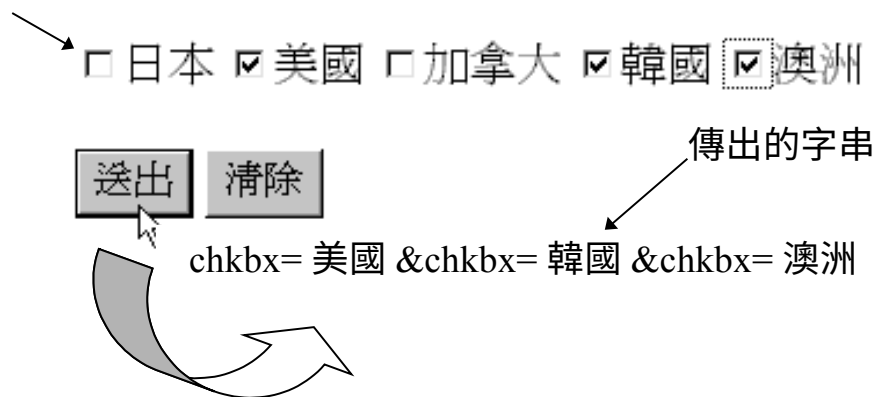
```
if (request.getParameter(" 核取方塊名稱 ") != null)  
{ ..... }
```

控制項的建立與訊息傳遞

- 核取方塊

- 群組核取方塊與選擇鈕，均是將屬於同一群組的控制項之名稱（**name 屬性**）設為相同。此時，核取方塊被選取後的傳出值，則不再是 **on** 字串，將是 **Value 屬性** 的值。
- 複選同一群組的核取方塊時，其傳出值將如同多選型清單方塊一般，將利用核取群組方塊的名稱，串連每個被選取控制項之傳出值。

核取方塊名稱為 chkbx



控制項的建立與訊息傳遞

- 利用隱藏控制項傳遞資料

- 撰寫網頁時，除了使用者填寫的資料外，常常會遇到需要傳遞額外資料的時候，但這些資料往往不是由使用者所填寫，所以並不希望被使用者看見。此時，可利用隱藏控制項，達到偷偷夾帶資料的目的。
- 以下為建立隱藏控制項的語法：

<Input Type = hidden Name= 控制項名稱 Value= 傳出值 >

- 各屬性的意義，如下表：

屬性	說明
Type	設定控制項的類型，建立隱藏控制項時，此屬性設為 hidden。
Name	控制項的名稱。
Value	此隱藏控制項的傳出值。

輸入資料的檢查

- 在那邊做資料檢查呢？

- 對於使用者輸入表單的資料，在更進一步利用前，應該先進行檢查 / 篩選資料的動作。

防止：

- 一．使用者輸入了錯誤的資料，導致系統的錯誤。
- 二．確認使用者所輸入的資料是否正確。

輸入資料的檢查

- 在那邊做資料檢查呢？

- 當然也可以在回應表單動作的 **JSP** 網頁中，執行檢查 / 篩選資料動作。
不過，把這個動作放在客戶端上執行，顯然是更好的主意；基本上，只要使用可在客戶端執行的網頁語言即可，**JavaScript** 就是其中一種。
- 客戶端與伺服器端執行資料檢查的優缺點：
 - 一．以相容性來看
 - 二．以執行效率來看
 - 三．以有效利用資源的角度來看

輸入資料的檢查

- 在客戶端執行資料檢查

- 在前端執行資料檢查最重要的物件是 **form** 物件。當在網頁定義一個表單後，便可利用 **form** 引用這個表單，至於表單的各個控制項，也都被包含於 **form** 物件。 **form** 物件包含於代表網頁之 **document** 物件的 **forms** 集合。
- 欲在客戶端執行資料檢查動作時，必須將原先執行表單資料送出的 **Submit** 按鈕以 **Button** 指令按鈕（ **type** 屬性設定為 **button** ）代替，然後定義一個回應該指令按鈕 **OnClick** 事件的函數