

B

jQuery表格與表單

- ◆ B-1 表單的基礎
- ◆ B-2 強化表單介面
- ◆ B-3 表單驗證
- ◆ B-4 美化表格內容
- ◆ B-5 分頁顯示表格

B-1 表單的基礎

表單是 HTML 網頁的互動介面，提供文字、密碼、選項按鈕、下拉式清單、核取方塊、多行文字方塊和按鈕等欄位來建立使用者介面。

B-1-1 HTML 表單標籤

HTML 表單是使用<form>標籤來建立，內含輸入資料或選項的欄位標籤，HTML 表單基本標籤的說明，如下表所示：

標籤	說明
<form> ... </form>	表單標籤
<input type=...>	表單輸入或選擇選項的欄位，包含按鈕和文字方塊欄位，不同 type 屬性是不同的欄位
<select> ... </select>	表單的選單欄位，擁有<option>標籤的選項
<option> ... </option>	選單欄位的選項
<textarea> ... </textarea>	多行文字方塊的備註欄位
<label>	搭配指定欄位的標題文字，使用 for 屬性指定
<button type=...>	按鈕欄位，type 屬性值可以是 button、submit 和 reset，它和<input>標籤建立按鈕的主要差異，在於可以建立圖片按鈕

HTML 表單是上表各種標籤的組合，其基本架構如下所示：

```
<form id="name" method="post | get"
      action="URL" enctype="MIME">
  <input type=...>
  <textarea> ... </textarea>
```

```
<select>
  <option> ... </option>
</select>
<input type="submit" ...>
</form>
```

上述<form>標籤可以建立擁有<input>、<textarea>和<select>欄位標籤的表單，在<select>標籤擁有<option>標籤的選項。<form>標籤的相關屬性說明，如下所示：

- id 屬性：表單名稱。
- method 屬性：設定資料傳送到伺服器端的方法，值 get 是使用 URL 網址的參數來傳遞；post 是使用 HTTP 通訊協定的標頭資料傳遞。
- action 屬性：設定伺服器端執行的表單處理程式，例如：CGI、ASP、ASP.NET、PHP 或 JSP 等程式檔案路徑。
- enctype 屬性：設定表單資料傳送的 MIME 型態，預設 application/x-www-form-urlencoded，如果 action 屬性值為電子郵件地址，表示使用電子郵件送出表單欄位內容，enctype 屬性 text/plain，表示是一般文字內容，如下所示：

```
<form action="mailto:hueyan@ms2.hinet.net" enctype="text/plain">
  ...
</form>
```

B-1-2 取得與指定表單元素值

在第 9-4-3 節已經說明如何使用 jQuery 選擇器來選擇表單元素，這一節筆者準備說明如何取得和指定表單欄位值，如下所示：

```
var price = $('#price').val();
var quantity = $('#quantity').val();
```

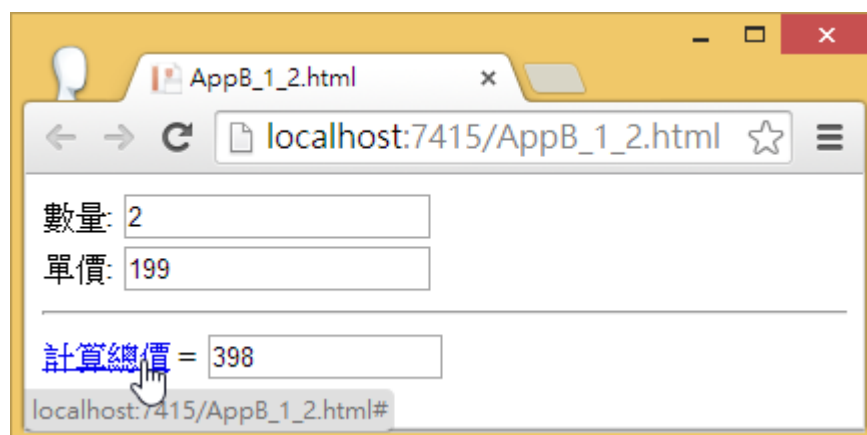
上述程式碼使用 `val()` 方法取得 `id` 屬性值 `price` 和 `quantity` 欄位的值，請注意！`val()` 方法沒有參數。指定表單欄位值也是使用 `val()` 方法，如下所示：

```
var amount = price * quantity;  
$('#amount').val(amount);
```

上述 `val()` 方法的參數就是指定的欄位值，以此例是指定 `id` 屬性值 `amount` 的欄位值。

jQuery 程式：AppB_1_2.html

在 jQuery 程式建立計算總價的表單，當輸入數量和單價後，按下方超連結，可以在等號之後的欄位顯示計算的總價，如下圖所示：



程式內容

```
01: <!DOCTYPE html>  
02: <html>  
03: <head>  
04: <meta charset="utf-8"/>  
05: <title>AppB_1_2.html</title>
```

```
06: <script src="jquery-2.1.1.min.js"></script>
07: <script>
08: $(document).ready(function() {
09:     $('#price').val(199);
10:     $('#a.cal').click(function() {
11:         var price = $('#price').val();
12:         var quantity = $('#quantity').val();
13:         var amount = price * quantity;
14:         $('#amount').val(amount);
15:         return false;
16:     });
17: });
18: </script>
19: </head>
20: <body>
21: <div class="content">
22:     <form method="post" id="calculate">
23:         <div>
24:             <label for="quantity" class="label">數量:</label>
25:             <input type="text" id="quantity" size="20">
26:         </div>
27:         <div>
28:             <label for="price" class="label">單價: </label>
29:             <input type="text" id="price" size="20">
30:         </div> <hr/>
31:         <a href="#" class="cal">計算總價</a> =
32:         <input type="text" id="amount" size="14">
33:     </form>
34: </div>
35: </body>
36: </html>
```

程式說明

- 第 9 列：指定第 29 列文字方塊的欄位初值。
- 第 10~16 列：第 31 列 a 元素的 click 事件處理，在第 11~12 列取得輸入的數量和總價，第 13 列計算總價，在第 14 列指定第 32 列文字方塊的欄位值，即總價。

- 第 22~33 列：HTML 表單擁有 3 個文字方塊的 input 和 label 元素。

B-1-3 判斷使用者輸入或選擇的欄位

HTML 表單的核取方塊欄位是使用 checked 屬性決定是否勾選，XHTML 需要指定屬性值 checked，如下所示：

```
<input type="checkbox" id="discount" checked="checked">
```

同樣的標籤在 HTML5 並不需要指定屬性值，只需加上 checked 屬性就表示勾選此核取方塊，如下所示：

```
<input type="checkbox" id="discount" checked>
```

在 jQuery 可以使用 attr() 方法替標籤加上 checked 屬性，如下所示：

```
$('#discount').attr('checked', true);
```

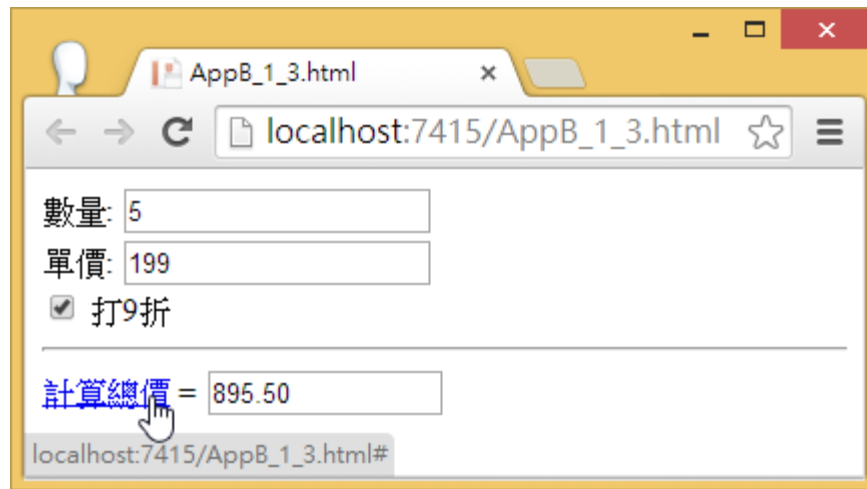
上述程式碼可以勾選 id 屬性值為 discount 的核取方塊。檢查核取方塊是否勾選也是使用 attr() 方法，只是沒有第 2 個參數，如下所示：

```
if ($('#discount').attr('checked')) {  
    amount = (price * quantity) * 0.9;  
    amount = amount.toFixed(2);  
}  
else {  
    amount = price * quantity;  
}
```

上述 if/else 條件判斷是否勾選 id 屬性值 discount 的核取方塊，如果是，就打 9 折，而且總價使用 toFixed() 方法四捨五入，只取出參數的小數點下 2 位。

jQuery 程式：AppB_1_3.html

這個 jQuery 程式是修改上一節的程式，新增打 9 折的核取方塊，勾選可以將總價打 9 折，如下圖所示：



上述表單因為勾選打 9 折，所以最後計算總價是 895.50，顯示到小數點下 2 位。

程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_1_3.html</title>
06: <script src="jquery-2.1.1.min.js"></script>
07: <script>
08: $(document).ready(function() {
09:     $('#price').val(199);
10:     $('#discount').attr('checked', true);
11:     $('#a.cal').click(function() {
12:         var price = $('#price').val();
13:         var quantity = $('#quantity').val();
14:         var amount;
15:         if ($('#discount').attr('checked')) {
16:             amount = (price * quantity) * 0.9;
17:             amount = amount.toFixed(2);
18:         }
19:         else {
20:             amount = price * quantity;
```

```
21:     }
22:     $('#amount').val(amount);
23:     return false;
24: });
25: });
26: </script>
27: </head>
28: <body>
29: <div class="content">
30:   <form method="post" id="calculate">
31:     <div>
32:       <label for="quantity" class="label">數量:</label>
33:       <input type="text" id="quantity" size="20">
34:     </div>
35:     <div>
36:       <label for="price" class="label">單價: </label>
37:       <input type="text" id="price" size="20">
38:     </div>
39:     <input type="checkbox" id="discount">
40:     <label for="discount">打 9 折</label><hr/>
41:     <a href="#" class="cal">計算總價</a> =
42:     <input type="text" id="amount" size="14">
43:   </form>
44: </div>
45: </body>
46: </html>
```

程式說明

- 第 10 列：指定第 39 列核取方塊為預設勾選，使用的是 `attr()` 方法。
- 第 15~21 列：if/else 條件檢查是否勾選核取方塊，如果有，在第 16 列打 9 折，沒有，執行第 20 列計算總價。
- 第 39 列：核取方塊的 HTML 標籤。

B-2 強化表單介面

我們可以活用之前學習的 jQuery 方法和事件處理來強化表單介面，讓 HTML 表單更有親和力，可以建立更人性化的表單使用介面。

B-2-1 讓第 1 個欄位取得焦點

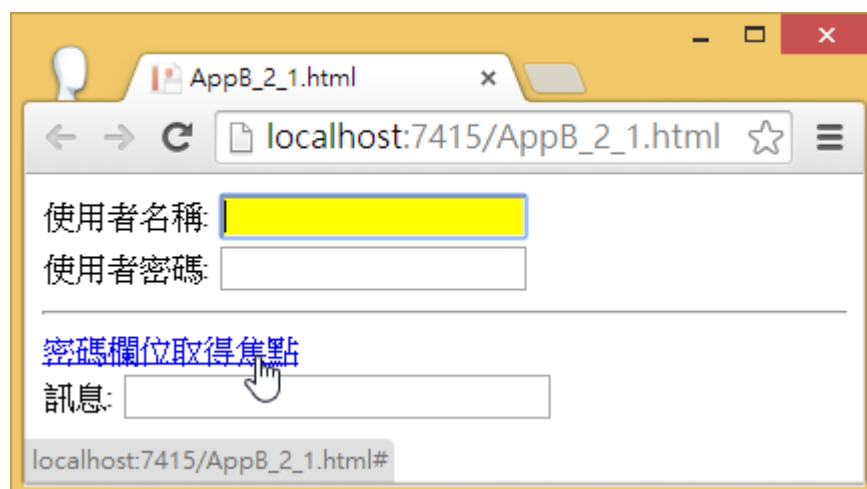
在表單輸入資料時，我們可以使用 `focus()` 方法讓指定欄位取得焦點，例如：在載入 HTML 表單網頁時，馬上讓該輸入資料的第 1 個欄位取得焦點，以方便使用者輸入資料，如下所示：

```
$('#login :text:first').focus().addClass('yellow');
```

上述程式碼是選擇 id 屬性值為 `login` 的表單，`:text` 表示之下的所有文字方塊，`:first` 是第 1 個，換句話說，就是表單第 1 個文字方塊取得焦點。

jQuery 程式：AppB_2_1.html

在 jQuery 程式建立簡單的登入表單，當第 1 次載入表單，可以看到第 1 個欄位取得焦點，其背景色彩是黃色，如下圖所示：



按下方超連結文字，可以讓第 2 個密碼欄位取得焦點。

程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_2_1.html</title>
06: <style type="text/css">
07: .yellow { background-color : yellow; }
08: </style>
09: <script src="jquery-2.1.1.min.js"></script>
10: <script>
11: $(document).ready(function() {
12:     $('#login :text:first').focus().addClass('yellow');
13:     $('a.pass').click(function() {
14:         $('#password').focus().addClass('yellow');
15:         return false;
16:     });
17:     $(':text').blur(function() {
18:         $(this).removeClass('yellow');
19:     });
20: });
21: </script>
22: </head>
23: <body>
24: <div class="content">
25:     <form method="post" id="login">
26:         <div>
27:             <label for="username">使用者名稱:</label>
28:             <input type="text" id="username" size="20">
29:         </div>
30:         <div>
31:             <label for="password">使用者密碼:</label>
32:             <input type="text" id="password" size="20">
33:         </div>
34:         <hr/>
35:         <a href="#" class="pass">密碼欄位取得焦點</a>
36:     </form>
37:     <form method="post" id="memo">
```

```
38:     <div>
39:         <label for="msg">訊息:</label>
40:         <input type="text" id="msg" size="30">
41:     </div>
42: </form>
43: </div>
44: </body>
45: </html>
```

程式說明

- 第 12 列：選擇表單第 1 個欄位後，呼叫 `focus()` 方法讓它取得焦點，`addClass()` 方法加上黃色背景。
- 第 13~16 列：第 35 列 `a` 元素的 `click` 事件處理，在第 14 列使用 `id` 屬性值選取第 32 列的密碼欄位後，使用 `focus()` 方法取得焦點，`addClass()` 方法加上黃色背景。
- 第 17~19 列：所有文字方塊的 `blur` 事件處理，當欄位失去焦點時，使用 `addClass()` 方法取消黃色背景。

B-2-2 取消或隱藏欄位

我們可以活用 `attr()` 方法和 `HTML` 屬性來取消表單欄位，即讓欄位沒有作用。不只如此，配合 jQuery 動畫方法，我們可以使用動畫方式來顯示或隱藏表單的額外欄位。

讓表單欄位沒有作用

`HTML` 標籤的 `disabled` 屬性可以決定欄位是可用或沒有作用，即是否允許輸入資料，例如：結帳時如果選現金，就不需要輸入信用卡資料，此時，就可以將相關欄位設為沒有作用，如下所示：

```
$('#creditCard input').attr('disabled', true);
```

上述程式碼使用 `attr()` 方法指定 `disabled` 屬性值為 `true`，表示這些欄位沒有作用。讓欄位有作用是將 `disabled` 屬性設為 `false`，如下所示：

```
$('#creditCard input').attr('disabled', false);
```

顯示或隱藏表單的額外欄位

在建立表單介面時，我們可以隱藏一些非必填欄位，只有當使用者需要輸入時，才顯示這些欄位，例如：在結帳時，如果帳單地址和送貨地址相同時，就不需顯示輸入送貨地址的欄位；只有當不同時，才需要勾選和顯示輸入送貨地址的額外欄位，如下所示：

```
$('#hideShip').click(function() {  
    if ($(this).attr('checked')) {  
        $('#shipping').slideUp('fast');  
    } else {  
        $('#shipping').slideDown('fast');  
    }  
});
```

上述程式碼是在 `id` 屬性值 `hideShip` 的核取方塊註冊 `click` 事件處理，`if/else` 條件判斷是否有勾選，有，就使用 `slideUp()` 方法隱藏表單欄位；反之，使用 `slideDown()` 方法顯示表單欄位。

jQuery 程式：AppB_2_2.html

在 jQuery 程式建立簡單的結帳表單，可以輸入帳單資訊和付款方式，預設是信用卡，所以下方信用卡欄位有作用，如果選【現金】，可以看到下方 2 個欄位變成沒有作用，如下圖所示：

帳單資訊

姓名:

地址:

付款方式: ☐ 信用卡 ☒ 現金 ☐ 轉帳

卡號:

有效期限:

送貨資訊

☒ 同帳單地址

上述表單最後是名為【同帳單地址】的核取方塊，如果帳單地址不同，取消勾選，就可以顯示輸入送貨資訊的欄位，如下圖所示：

帳單資訊

姓名:

地址:

付款方式: ☐ 信用卡 ☒ 現金 ☐ 轉帳

卡號:

有效期限:

送貨資訊

☐ 同帳單地址

姓名:

地址:

程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_2_2.html</title>
06: <script src="jquery-1.8.3.min.js"></script>
07: <script>
08: $(document).ready(function() {
09:     $('#order :text:first').focus();
10:     $('#credit').attr('checked', true);
11:     $('#hideShip').attr('checked', true);
12:     $('#shipping').hide();
13:     // 輸入信用卡欄位不可用
14:     $('#transfer, #cash').click(function() {
15:         $('#creditCard input').attr('disabled', true)
16:         .css('backgroundColor', '#CCC');
17:         $('#creditCard label').css('color', '#BBB');
18:     });
19:     // 輸入信用卡欄位可用
20:     $('#credit').click(function() {
21:         $('#creditCard input').attr('disabled', false)
22:         .css('backgroundColor', '');
23:         $('#creditCard label').css('color', '');
24:         $('#cardNumber').focus();
25:     });
26:     // 隱藏與顯示送貨資訊欄位
27:     $('#hideShip').click(function() {
28:         if ($(this).attr('checked')) {
29:             $('#shipping').slideUp('fast');
30:             $('#shipping').hide();
31:         } else {
32:             $('#shipping').show();
33:             $('#shipping').slideDown('fast');
34:         }
35:     });
36: });
37: </script>
38: </head>
39: <body>
40: <div class="content">
```

```

41: <form action="order.php" method="post" id="order">
42:   <fieldset>
43:     <legend>帳單資訊</legend>
44:     <div><label for="name">姓名:</label>
45:       <input type="text" id="name" size="20"></div>
46:     <div><label for="address">地址: </label>
47:       <input type="text" id="address" size="20"></div>
48:     <div><span>付款方式: </span>
49:       <input type="radio" name="payment" id="credit"
50:         value="credit">
51:       <label for="credit">信用卡</label>
52:       <input type="radio" name="payment" id="cash"
53:         value="cash">
54:       <label for="cash">現金</label>
55:       <input type="radio" name="payment" id="transfer"
56:         value="transfer">
57:       <label for="paypal">轉帳</label>
58:     </div>
59:     <div id="creditCard">
60:       <div><label for="cardNumber">卡號: </label>
61:         <input type="text" id="cardNumber"></div>
62:       <div><label for="expiration">有效期限: </label>
63:         <input type="text" id="expiration"></div>
64:     </div>
65:   </fieldset>
66:   <fieldset>
67:     <legend>送貨資訊</legend>
68:     <div><input type="checkbox" id="hideShip">
69:       <label for="hideShip">同帳單地址</label></div>
70:     <div id="shipping">
71:       <div><label for="shipName">姓名: </label>
72:         <input type="text" id="shipName" size="20"></div>
73:       <div><label for="shipAddress">地址: </label>
74:         <input type="text" id="shipAddress" size="20"></div>
75:     </div>
76:   </fieldset>
77: </form>
78: </div>
79: </body>

```

程式說明

- 第 6 列：經筆者測試 jQuery 2.1.1 版並無法顯示送貨資訊，所以本節範例是使用 1.8.3 版。
- 第 9 列：讓第 45 列的第 1 個欄位取得焦點。
- 第 10 列：選擇第 49~50 列的選擇按鈕。
- 第 11 列：核取第 68 列的核取方塊。
- 第 12 列：隱藏第 70~75 列的 div 元素，即輸入送貨資訊的欄位。
- 第 14~18 列：第 52~53 列和第 55~56 列☐標籤的 click 事件處理，在第 15~16 列使用 attr()方法將第 61 列和第 63 列的 input 元素設為沒有作用，並且使用 css()方法更改其背景色彩，第 17 列指定第 60 列和第 62 列<label>元素的色彩。
- 第 20~25 列：第 49~51 列☐標籤的 click 事件處理，在第 21~22 列使用 attr()方法將第 61 列和第 63 列的<input>元素設為有作用，並且使用 css()方法取消背景色彩，第 23 列取消第 60 和 62 列 label 元素的色彩，在第 24 列讓第 61 列的欄位取得焦點。
- 第 27~35 列：第 68 列☐標籤的 click 事件處理，在第 28~34 列的 if/else 條件判斷是否勾選，如果勾選，在第 29~30 列使用 slideUp()和 hide()方法隱藏第 70~75 列的表單欄位，沒有勾選，在第 32~33 列使用 slideDown()和 show()方法顯示第 70~75 列的表單欄位。
- 第 41~77 列：HTML 表單標籤，使用 legend 元素分成兩大部分，即帳單資訊和送貨資訊。

B-2-3 建立表單欄位的提示說明

表單欄位標籤的 value 屬性值是欄位預設值，我們可以將它建立成欄位的提示文字，提示欄位輸入哪一些資訊，如下所示：

```
<input type="text" id="name" value="姓名"/>
```

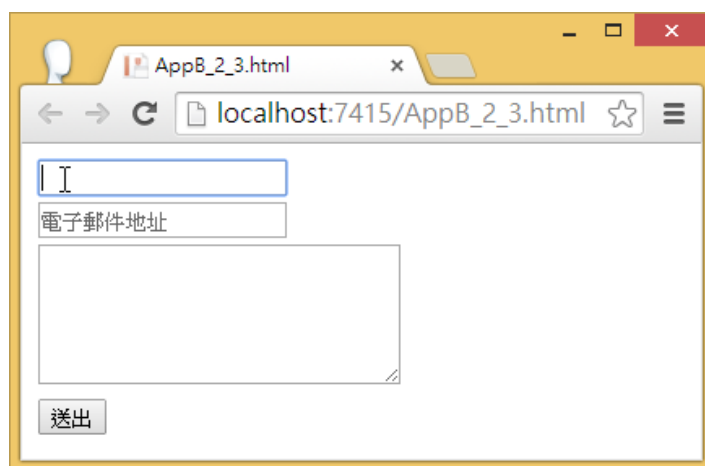
上述文字方塊顯示預設值"姓名"，我們準備使用 jQuery 程式碼替所有標籤的 value 屬性值建立成表單欄位提示，當欄位取得焦點時清除欄位內容；失去焦點且沒有輸入資料時，顯示 value 屬性值，如下所示：

```
$('.text').each(function() {  
    var hint = $(this).val();  
    $(this).focus(function() {  
        if($(this).val() == hint || "") {  
            $(this).val(""); // 清除  
        }  
    }).blur(function() {  
        if($(this).val() == "") {  
            $(this).val(hint); // 顯示  
        }  
    });  
});
```

上述 each()方法可以處理每一個標籤，變數 hint 是 value 屬性值，在 focus 事件處理使用 if 條件判斷欄位值是否是提示說明或空字串，如果是，就清除欄位內容；blur 事件處理則是使用 if 條件判斷是否有輸入資料，如果沒有資料，就顯示 value 屬性值的提示說明文字。

jQuery 程式：AppB_2_3.html

在 jQuery 程式表單的所有標籤，將 value 屬性值建立成表單欄位提示，當欄位取得焦點就清除提示說明，如下圖所示：



上述表單前 2 個欄位的灰色字是提示說明，因為第 1 個欄位取得焦點，所以清除原來的提示說明"姓名"。

程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_2_3.html</title>
06: <style type="text/css">
07: .inactive { color: #666666; }
08: </style>
09: <script src="jquery-2.1.1.min.js"></script>
10: <script>
11: $(document).ready(function() {
12:     $(':text').each(function() {
13:         var hint = $(this).val();
14:         $(this).addClass('inactive')
15:         .focus(function() {
16:             $(this).removeClass('inactive');
17:             if($(this).val() == hint || "") {
18:                 $(this).val(""); // 清除
19:             }
20:         }).blur(function() {
21:             if($(this).val() == "") {
```

```
22:         $(this).addClass('inactive');
23:         $(this).val(hint); // 顯示
24:     }
25: });
26: });
27: });
28: </script>
29: </head>
30: <body>
31: <div class="content">
32:   <form action="">
33:     <input type="text" id="name" value="姓名"/><br/>
34:     <input type="text" id="email" value="電子郵件地址"/>
35:     <br/><textarea rows="5" cols="25" id="comment">
36:       </textarea><br/>
37:     <input type="submit" value="送出"/>
38:   </form>
39: </div>
40: </body>
41: </html>
```

程式說明

- 第 12~26 列：選出所有<input type="text">標籤後，使用 each()方法一一取出各元素來呼叫匿名函數。
- 第 13 列：變數 hint 是文字方塊的 value 屬性值。
- 第 14 列：在<input type="text">標籤套用提示說明的樣式類別 inactive。
- 第 15~20 列：註冊 focus 事件處理，在第 16 列移除 inactive 樣式類別，第 17~19 列的 if 條件判斷欄位值是否是提示說明或空字串，如果是，在第 18 列使用 val()方法清除欄位內容。

- 第 20~25 列：註冊 blur 事件處理，在第 21~24 列的 if 條件判斷欄位值是否是空字串，如果是，第 22 列套用提示說明的樣式類別 inactive，在第 23 列使用 val()方法指定欄位內容是提示說明。

B-2-4 全選

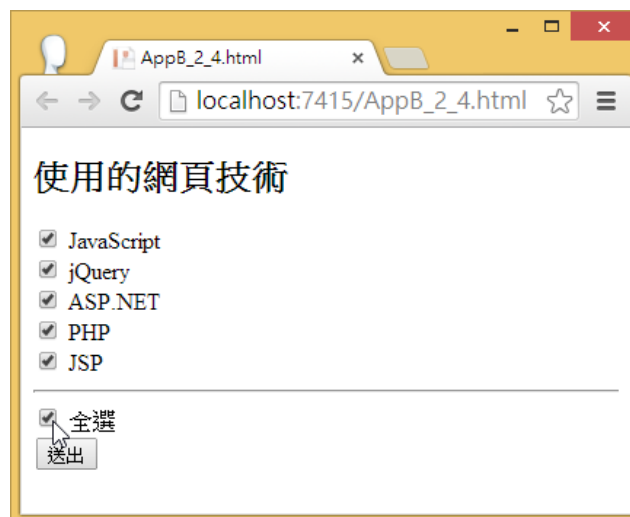
全選功能是當表單擁有多個核取方塊時，為了方便勾選，只需勾選名為【全選】的核取方塊，就可以一次勾選全部核取方塊，如下所示：

```
$('.checkAll:checkbox').change(function() {  
    $(':checkbox[id="web"]').attr('checked', true);  
});
```

上述程式碼在 class 屬性值為 checkAll 的核取方塊註冊 change 事件處理，當勾選，就將 id 屬性值為"web"的核取方塊，使用 attr()方法指定 checked 屬性為 true，即全部勾選。

jQuery 程式：AppB_2_4.html

在 jQuery 程式的表單新增多個核取方塊，並且在最下方建立名為【全選】的核取方塊，如下圖所示：



勾選【全選】核取方塊，就可以同時勾選上方清單的全部核取方塊。

程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_2_4.html</title>
06: <script src="jquery-2.1.1.min.js"></script>
07: <script>
08: $(document).ready(function() {
09:     $('.checkAll:checkbox').change(function() {
10:         $(':checkbox[id="web"]').attr('checked', true);
11:     });
12: });
13: </script>
14: </head>
15: <body>
16: <div class="content">
17:     <form action="">
18:         <div class="stats">
19:             <h2 class="title">使用的網頁技術</h2>
20:             <input type="checkbox" id="web" value="JavaScript"/>
21:             JavaScript<br/>
22:             <input type="checkbox" id="web" value="jQuery"/>
23:             jQuery<br/>
24:             <input type="checkbox" id="web" value="ASP.NET"/>
25:             ASP.NET<br/>
26:             <input type="checkbox" id="web" value="PHP"/>
27:             PHP<br/>
28:             <input type="checkbox" id="web" value="JSP"/>
29:             JSP<br/>
30:             <hr/>
31:             <input type="checkbox" class="checkAll" id="web" />
32:             <span>全選</span><br/>
33:             <input type="submit" value="送出"/>
34:         </div>
35:     </form>
36: </div>
```

```
37: </body>
38: </html>
```

程式說明

- 第 9~11 列：將 id 屬性值為"web"的核取方塊，使用 attr()方法指定 checked 屬性為 true，即全部勾選。

B-3 表單驗證

表單驗證（Form Validation）就是驗證使用者在表單欄位輸入的資料是否正確，對於網頁設計來說，這是一件十分重要的工作。

因為使用者常常可能輸錯資料，例如：忘了輸入、資料範圍錯誤或格式不正確，輸入錯誤資料，輕者只是顯示錯誤；嚴重時可能影響整個 Web 應用程式的執行。

B-3-1 submit 事件的表單驗證

當使用者在表單按下標籤的按鈕，就表示送出表單，此時會觸發 submit 事件，如下所示：

```
$("#form").submit( function() {
    ...
    return true;
});
```

上述 submit()方法註冊表單的 submit 事件處理，在匿名函數使用 return 關鍵字傳回 true，表示送出表單；傳回 false，表示表單不會送出。在 submit 事件處理的匿名函數可以檢查使用者輸入的欄位資料，執行表單驗證，以便決定是否送出表單，如下所示：

```
var hasError = false;
$(this).find(":text").each(function(){
    if ($(this).val().length == 0) {
        alert("錯誤: 文字方塊需要輸入值!");
    }
});
```

```
$(this).focus();  
hasError = true;  
return false;  
}  
});  
if (hasError) return false;
```

上述程式碼可以找出所有標籤，然後使用 `each()` 方法一一檢查各文字方塊欄位輸入的資料，`if` 條件判斷是否有輸入資料，沒有，字串長度 `length` 屬性值為 0，就使用 `alert()` 函數顯示錯誤的訊息視窗。

當表單有欄位忘了輸入資料，就會將變數 `hasError` 指定成 `true`，所以最後可以使用 `if` 條件檢查是否有錯誤，如果有，就傳回 `false` 不送出表單。

jQuery 程式：AppB_3_1.html

在 jQuery 程式的表單新增 `submit` 事件的表單驗證，可以檢查文字方塊欄位是否忘了輸入資料，當按【送出】鈕，如果有空欄位，就顯示錯誤的訊息視窗，如下圖所示：



程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_3_1.html</title>
06: <script src="jquery-2.1.1.min.js"></script>
07: <script>
08: $(document).ready(function() {
09:     $("form").submit( function() {
10:         var hasError = false;
11:         $(this).find(":text").each(function() {
12:             if ($(this).val().length == 0) {
13:                 alert("錯誤: 文字方塊需要輸入值!");
14:                 $(this).focus();
15:                 hasError = true;
16:                 return false;
17:             }
18:         });
19:         if (hasError) return false;
20:         return true;
21:     });
22: });
23: </script>
24: </head>
25: <body>
26: <div class="content">
27:     <form action="">
28:         姓名: <input type="text" id="name"/><br/>
29:         電郵: <input type="text" id="email"/><br/>
30:         <textarea rows="5" cols="25" id="comment">
31:         </textarea><br/>
32:         <input type="submit" value="送出"/>
33:     </form>
34: </div>
35: </body>
36: </html>
```

程式說明

- 第 9~21 列：在第 27~33 列的 form 元素註冊 submit 事件處理，在第 10 列的 hasError 變數判斷表單欄位是否有錯誤。
- 第 11~18 列：選出所有<input type="text">標籤的文字方塊後，使用 each()方法一一取出文字方塊來呼叫匿名函數，在第 12~17 列的 if 條件判斷文字方塊是否有輸入資料，如果沒有，第 13 列顯示訊息視窗，在第 14 列取得焦點，第 15 列將變數 hasError 指定成 true，表示有錯誤，最後在第 16 列傳回 false，return 關鍵字可以跳出 each()方法的匿名函數。
- 第 19 列：if 條件判斷 hasError 變數是否是 true，即有錯誤，如果有，就傳回 false，不送出表單。

B-3-2 click 事件的表單驗證

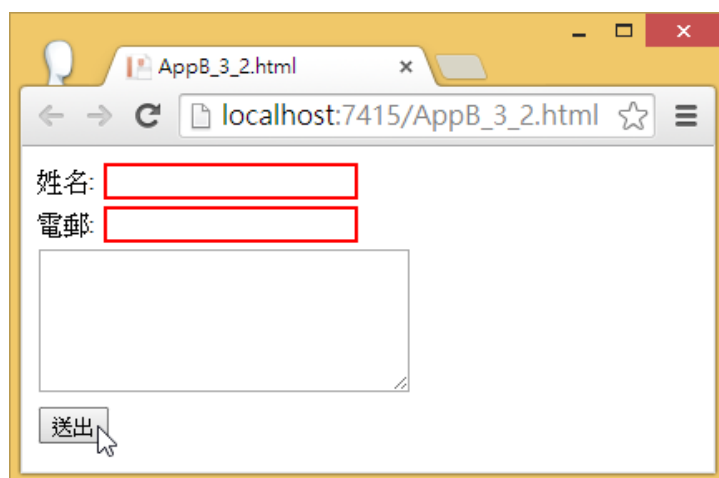
除了使用 submit 事件進行表單驗證外，另一種比較好的方法是註冊<input type="submit">標籤的 click 事件處理來執行表單驗證，如下所示：

```
$(':submit').click(function(event) {  
    $(':text').each(function() {  
        if($(this).val().length == 0) {  
            $(this).addClass('error');  
        }  
    });  
    event.preventDefault();  
});
```

上述程式碼使用:submit 選擇<input type="submit">標籤後，註冊 click 事件處理，然後找出所有<input type="text">標籤，使用 each()方法一一檢查各文字方塊欄位輸入的資料，if 條件判斷是否有輸入資料，沒有，字串長度 length 屬性值為 0，就使用 addClass()方法在欄位顯示紅色框線。

jQuery 程式：AppB_3_2.html

在 jQuery 程式的表單新增 click 事件的表單驗證，可以檢查文字方塊欄位是否忘了輸入資料，當按【送出】鈕，如果有空欄位，就在文字方塊顯示紅色框線，如下圖所示：



程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_3_2.html</title>
06: <style type="text/css">
07: .error { border: 2px solid red; }
08: </style>
09: <script src="jquery-2.1.1.min.js"></script>
10: <script>
11: $(document).ready(function() {
12:     $(':submit').click(function(event) {
13:         $(':text').each(function() {
14:             if($(this).val().length == 0) {
15:                 $(this).addClass('error');
```

```
16:         }
17:     });
18:     event.preventDefault();
19: });
20: $(':input').focus(function() {
21:     $(this).removeClass('error');
22: });
23: });
24: </script>
25: </head>
26: <body>
27: <div class="content">
28:     <form action="">
29:         姓名: <input type="text" id="name"/><br/>
30:         電郵: <input type="text" id="email"/><br/>
31:         <textarea rows="5" cols="25" id="comment">
32:         </textarea><br/>
33:         <input type="submit" value="送出"/>
34:     </form>
35: </div>
36: </body>
37: </html>
```

程式說明

- 第 12~19 列：在第 33 列的<input type="submit">標籤註冊 click 事件處理，在第 13~17 列選出所有文字方塊後，使用 each()方法一一取出文字方塊來呼叫匿名函數，在第 14~16 列的 if 條件判斷文字方塊是否有輸入資料，如果沒有，第 15 列套用 error 樣式類別，也就是在文字方塊外顯示紅色框線。
- 第 20~22 列：使用:input 選擇輸入元素來註冊 focus 事件處理，在第 21 列移除 error 樣式類別。

B-3-3 即時驗證欄位資料

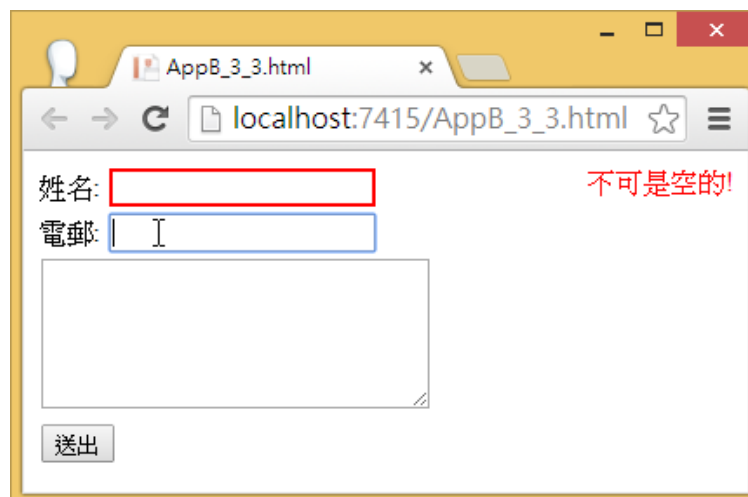
在前 2 節的表單驗證都需要按下送出鈕後，才能執行表單驗證，並不能即時提供使用者輸入資料是否正確的訊息，為了能夠即時驗證欄位資料，我們可以在 `blur` 事件處理執行欄位資料的檢查，如下所示：

```
$(':input').blur(function() {  
    if($(this).val().length == 0) {  
        $(this).addClass('error')  
        .after('<span class="error">不可是空的!</span>');  
    }  
});
```

上述程式碼使用 `:input` 選擇輸入元素來註冊 `blur` 事件處理，當欄位失去焦點時，馬上使用 `if` 條件判斷是否有輸入資料，如果沒有，就在之後新增紅色錯誤訊息的 `span` 元素。

jQuery 程式：AppB_3_3.html

在 jQuery 程式的表單新增 `blur` 事件的表單驗證，當欄位失去焦點就檢查文字方塊欄位是否忘了輸入資料，如下圖所示：



上述圖例在選擇第 1 個欄位取得焦點後，再按第 2 個欄位，就可以讓第 1 個欄位失去焦點，因為沒有輸入資料，所以在後面顯示紅色的錯誤訊息。

程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_3_3.html</title>
06: <style type="text/css">
07: span.error { float: right; color : red; }
08: input.error { border: 2px solid red; }
09: </style>
10: <script src="jquery-2.1.1.min.js"></script>
11: <script>
12: $(document).ready(function() {
13:     $(':input').blur(function() {
14:         if($(this).val().length == 0) {
15:             $(this).addClass('error')
16:             .after('<span class="error">不可是空的!</span>');
17:         }
18:     });
19:     $(':input').focus(function() {
20:         $(this).removeClass('error')
21:         .next('span').remove();
22:     });
23: });
24: </script>
25: </head>
26: <body>
27: <div class="content">
28:     <form action="">
29:         姓名: <input type="text" id="name"/><br/>
30:         電郵: <input type="text" id="email"/><br/>
31:         <textarea rows="5" cols="25" id="comment">
32:         </textarea><br/>
33:         <input type="submit" value="送出"/>
34:     </form>
```

```
35: </div>
36: </body>
37: </html>
```

程式說明

- 第 13~18 列：使用 選擇輸入元素來註冊 blur 事件處理，在第 14~17 列的 if 條件判斷文字方塊是否有輸入資料，如果沒有，第 15 列套用 error 樣式類別，也就是在文字方塊外顯示紅色框線，在第 16 列新增 span 元素的錯誤訊息文字。
- 第 19~22 列：使用 選擇輸入元素來註冊 focus 事件處理，在第 20 列移除 error 樣式類別，第 21 列刪除 span 元素。

B-4 美化表格內容

jQuery 方法可以輕鬆替表格加上特效，在之前已經說明過斑馬紋效果，這一節將說明折疊表格列、選取資料列的 Hover 特效和替特殊儲存格內容加上提示文字，不只如此，我們還可以編輯表格列，在表格新增和刪除表格列。

B-4-1 折疊表格列的滑動特效

jQuery 可以輕鬆替表格的表格列建立折疊表格列的滑動特效，如下所示：

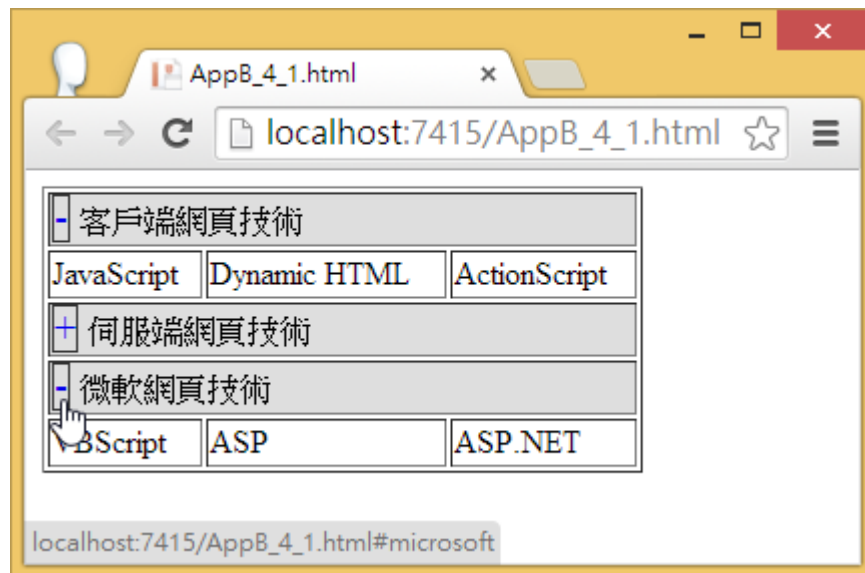
```
$('#a').click(function() {
    var id = $(this).attr('href');
    if ( $(id).css('display') == 'none' ) {
        $(id).slideDown('slow');
        $(this).html('-');
    } else {
        $(id).slideUp('slow');
        $(this).html('+');
    }
    return false;
});
```

```
});
```

上述程式碼註冊 a 元素的 click 事件處理，變數 id 取得點選 a 元素的 href 屬性值，它是折疊目標 tr 元素的 id 屬性值，if/else 條件使用 css() 方法取得 display 樣式屬性值來檢查 tr 元素是顯示或隱藏，如果隱藏就是使用 slideDown() 方法顯示表格列，反之，隱藏表格列。

jQuery 程式：AppB_4_1.html

在 jQuery 程式新增 HTML 表格後，建立折疊表格列的滑動特效，如下圖所示：



按一下表格列前方的「+」號，可以展開下一列表格列；按一下「-」號就會折疊表格列。

程式內容

```
01: <!DOCTYPE html>
```

```
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_4_1.html</title>
06: <style type="text/css">
07: a { text-decoration : none;
08:     font-size : 20px;
09:     border : 1px solid #333;  }
10: .title { background-color : #DEDEDE; }
11: </style>
12: <script src="jquery-2.1.1.min.js"></script>
13: <script>
14: $(document).ready(function() {
15:     $('tr:even').addClass('title');
16:     $('tr:odd').hide();
17:     $('a').click(function() {
18:         var id = $(this).attr('href');
19:         if ( $(id).css('display') == 'none' ) {
20:             $(id).slideDown('slow');
21:             $(this).html('-');
22:         } else {
23:             $(id).slideUp('slow');
24:             $(this).html('+');
25:         }
26:         return false;
27:     });
28: });
29: </script>
30: </head>
31: <body>
32: <table border="1" style="width: 300px">
33: <tr>
34:     <td colspan="3"><a href="#client">+</a>
35:     客戶端網頁技術</td>
36: </tr>
37: <tr id="client">
38:     <td>JavaScript</td>
39:     <td>Dynamic HTML</td>
40:     <td>ActionScript</td>
41: </tr>
42: <tr>
43:     <td colspan="3"><a href="#server">+</a>
44:     伺服器端網頁技術</td>
```



```

45: </tr>
46: <tr id="server">
47:   <td>JSP</td>
48:   <td>PHP</td>
49:   <td>ASP</td>
50: </tr>
51: </div>
52: <tr>
53:   <td colspan="3"><a href="#microsoft">+</a>
54:   微軟網頁技術</td>
55: </tr>
56: <tr id="microsoft">
57:   <td>VBScript</td>
58:   <td>ASP</td>
59:   <td>ASP.NET</td>
60: </tr>
61: </table>
62: </body>
63: </html>

```

程式說明

- 第 15 列：在表格的偶數列套用第 10 列的 `title` 樣式類別。
- 第 16 列：隱藏表格的奇數列，即第 37~41 列、第 46~50 列和第 56~60 列的表格列。
- 第 17~27 列：在 `a` 元素註冊 `click` 事件處理，在第 18 列取得 `a` 元素的 `href` 屬性值，即目標的 `id` 屬性值，第 19~25 列的 `if/else` 條件使用 `css()` 方法取出 `id` 屬性值為 `id` 變數值的 `display` 樣式屬性，如果為 `none`，表示是隱藏，所以在第 20~21 列使用 `slideDown()` 方法來顯示表格列，並且更改內容為減號，第 23~24 列是顯示，所以使用 `slideUp()` 方法隱藏表格列，並且更改內容為加號。
- 第 34、43 和 53 列：`a` 元素的 `href` 屬性值是分別對應第 37、46 和 56 列 `tr` 元素的 `id` 屬性值。

B-4-2 在表格列加上 **Hover** 特效

在表格列可以使用 `hover` 事件加上特效來增加互動功能，例如：當滑鼠游標進入表格列就顯示不同的背景色彩；離開恢復成原來色彩，如下所示：

```
$(tr).hover(  
    function() {  
        $(this).addClass('active')  
    }, function() {  
        $(this).removeClass('active');  
    });
```

上述程式碼是在表格列 `tr` 元素註冊 `hover` 事件處理，進入使用 `addClass()` 方法套用 `active` 樣式類別；離開就移除此樣式類別。

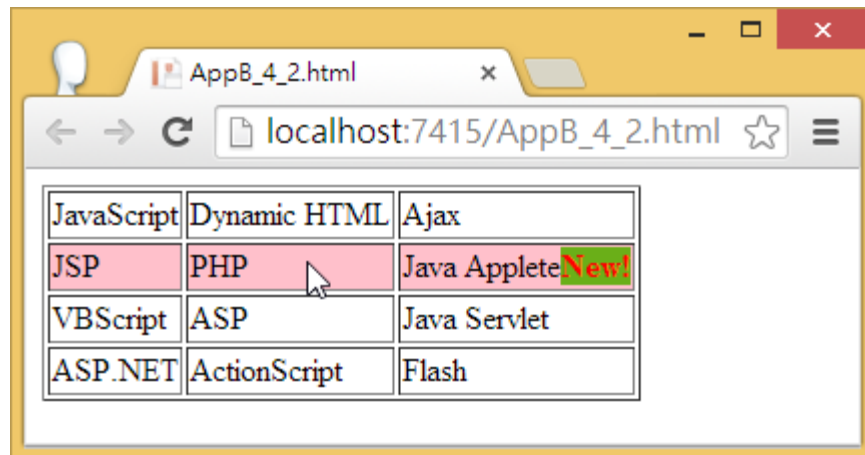
不只如此，我們還可以找出包含特定內容的儲存格，在儲存格新增所需的 **HTML** 元素，如下所示：

```
$(this).find('td:contains("Java")')  
    .append('<span class="special">New!</span>');
```

上述程式碼使用 `find()` 方法找尋參數的 `td` 元素，`:contains()` 可以找出符合指定字串的元素，以此例就是包含 `"Java"` 字串的儲存格，找到後，使用 `append()` 方法在儲存格新增 `span` 元素的特殊字串，以此例是 `New!`。

jQuery 程式：AppB_4_2.html

在 jQuery 程式建立 **HTML** 表格後，使用 `hover` 事件來建立特效，如下圖所示：



上述圖例當滑鼠游標進入表格列就顯示粉紅色的背景色彩，因為此列的最後一個儲存格包含 Java 字串，所以新增 New!字串。

程式內容

```

01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_4_2.html</title>
06: <style type="text/css">
07: .active { background-color: pink; }
08: .special { background : #6AAF18;
09:             text-align : center;
10:             color : red;
11:             font-weight : bold;}
12: </style>
13: <script src="jquery-2.1.1.min.js"></script>
14: <script>
15: $(document).ready(function() {
16:     $('tr').hover(
17:         function() {
18:             $(this).addClass('active')
19:             $(this).find('td:contains("Java")')
20:                 .append('<span class="special">New!</span>');
21:         }, function() {

```

```
22:         $(this).removeClass('active');
23:         $('.special').remove();
24:     });
25: });
26: </script>
27: </head>
28: <body>
29: <table border="1">
30: <tr>
31:     <td>JavaScript</td>
32:     <td>Dynamic HTML</td>
33:     <td>Ajax</td>
34: </tr><tr>
35:     <td>JSP</td>
36:     <td>PHP</td>
37:     <td>Java Applete</td>
38: </tr><tr>
39:     <td>VBScript</td>
40:     <td>ASP</td>
41:     <td>Java Servlet</td>
42: </tr><tr>
43:     <td>ASP.NET</td>
44:     <td>ActionScript</td>
45:     <td>Flash</td>
46: </tr>
47: </table>
48: </body>
49: </html>
```

程式說明

- 第 16~24 列：第 30~46 列 tr 元素的 hover 事件處理，第 17~21 列是第 1 個匿名函數；第 21~24 列是第 2 個匿名函數。
- 第 17~21 列：當滑鼠游標進入表格列就執行第 1 個匿名函數，在第 18 列使用 addClass()方法套用第 7 列的 CSS 樣式類別 active，第 19~20 列使用 find()方法找尋包含"Java"字串的 td 元素，在第 20 列使

用 `append()` 方法新增 `span` 元素，此元素套用第 8~11 列的 CSS 樣式類別 `special`。

- 第 21~24 列：當滑鼠游標離開表格列就執行第 2 個匿名函數，在第 22 列使用 `removeClass()` 方法移除 CSS 樣式類別 `active`，第 23 列使用 `remove()` 方法刪除第 20 列新增的 `span` 元素。

B-4-3 表格列的操作

jQuery 提供更改 DOM 的相關方法，我們可以使用這些方法來執行表格列的相關操作，即在表格刪除和插入表格列。

刪除表格列

在表格刪除表格列是使用 `remove()` 方法，我們只需選擇指定的表格列，就可以刪除選擇的表格列，如下所示：

```
$(tr:eq(1)).remove();  
$(tr:last).remove();
```

上述程式碼刪除 `eq(1)` 的 `tr` 元素，即索引值為 1 的表格列（索引值是從 0 開始），即第 2 列表格列，`:last` 是最後一個表格列。

插入表格列

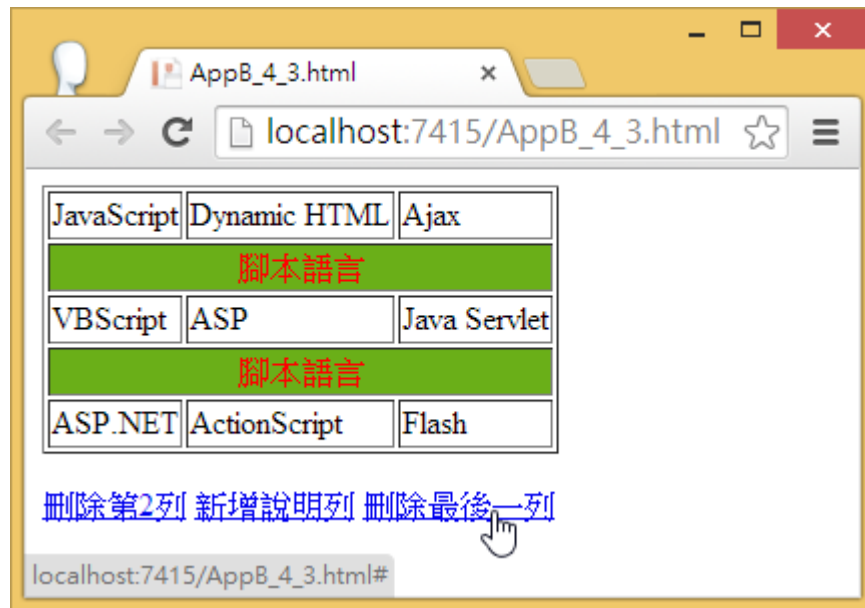
在表格插入表格列可以使用 `before()` 和 `after()` 方法，例如：在之後插入一個新的表格列，如下所示：

```
$(tr:contains("Script"))  
.after('<tr><td colspan="3" ' +  
      'class="special">腳本語言</td></tr>');
```

上述程式碼在找出 `tr` 元素內容包含 "Script" 的表格列之後，插入一個新的表格列。

jQuery 程式：AppB_4_3.html

在 jQuery 程式建立表格列和下方的 3 個超連結，按下超連結可以刪除表格列，和在包含"Script"的表格列之後新增說明列，如下圖所示：



程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>AppB_4_3.html</title>
06: <style type="text/css">
07: .special { background : #6AAF18;
08:             text-align : center;
09:             color : red; }
10: </style>
11: <script src="jquery-2.1.1.min.js"></script>
12: <script>
13: $(document).ready(function() {
```

```
14:    $('a.select1').click(function() {
15:        $('tr:eq(1)').remove();
16:        return false;
17:    });
18:    $('a.select2').click(function() {
19:        $('tr:contains("Script")')
20:        .after('<tr><td colspan="3" ' +
21:            'class="special">腳本語言</td></tr>');
22:        return false;
23:    });
24:    $('a.select3').click(function() {
25:        $('tr:last').remove();
26:        return false;
27:    });
28: });
29: </script>
30: </head>
31: <body>
32: <table border="1">
33: <tr>
34:     <td>JavaScript</td>
35:     <td>Dynamic HTML</td>
36:     <td>Ajax</td>
37: </tr><tr>
38:     <td>JSP</td>
39:     <td>PHP</td>
40:     <td>Java Applete</td>
41: </tr><tr>
42:     <td>VBScript</td>
43:     <td>ASP</td>
44:     <td>Java Servlet</td>
45: </tr><tr>
46:     <td>ASP.NET</td>
47:     <td>ActionScript</td>
48:     <td>Flash</td>
49: </tr>
50: </table>
51: <p><a href="#" class="select1">刪除第 2 列</a>
52: <a href="#" class="select2">新增說明列</a>
53: <a href="#" class="select3">刪除最後一列</a></p>
54: </body>
55: </html>
```

程式說明

- 第 14~17 列：在第 51 列的 a 元素註冊 click 事件處理，第 15 列刪除第 2 列表格列。
- 第 18~23 列：在第 52 列的 a 元素註冊 click 事件處理，第 19~21 列是在包含"Script"的表格列之後新增說明列。
- 第 24~27 列：在第 53 列的 a 元素註冊 click 事件處理，第 25 列刪除最後一列表格列。

B-5 分頁顯示表格

如果 HTML 表格很長時，受限於螢幕解析度，我們無法在瀏覽器的同一頁網頁顯示整個表格內容，此時，可以使用分頁方式來顯示部分表格，並且提供超鏈結來切換顯示其他頁面的表格。

計算分頁數

在使用 jQuery 建立分頁顯示表格前，我們需要先計算表格可以分成幾頁，首先定義每頁有幾列的 `rowPage` 變數，以此例每頁有 3 列，和取得表格共有幾列，如下所示：

```
var rowsPage = 3;  
var totalRows = $('#books tr').length;
```

上述第 2 列程式碼使用 jQuery 選擇器選出所有表格列後，使用 `length` 屬性取得共有幾列。然後計算表格列可以分成幾頁，如下所示：

```
var numPages = Math.floor(totalRows/rowsPage);  
var mod = totalRows % rowsPage;  
if (mod > 0) numPages++;
```


上述程式碼使用 `totalRows / rowsPage` 除法計算頁數，`Math.floor()` 方法可以傳回最大整數，因為如果不能整除會少一頁，所以使用餘數判斷是否可以整除，`if` 條件判斷餘數是否大於 0，大於 0 表示不能整除，所以頁數多一頁。

顯示頁碼超連結

在計算出分頁數後，就可以建立頁碼超連結，使用的是 `for` 迴圈，如下所示：

```
for ( i = 0; i < numPages; i++ ) {  
    var pageNum = i + 1;  
    $('#nav').append('<a href="#" rel="' +  
        i + '">' + pageNum + '</a> ');  
}
```

上述 `for` 迴圈是從分頁索引 0 至 `numPages-1`，在取得頁碼編號，即索引加一後，在 `id` 屬性值 `nav` 的 `div` 元素新增頁碼超連結的 `a` 元素，`rel` 屬性值就是分頁索引。

顯示表格的第 1 頁分頁

現在，我們就可以顯示第 1 頁的表格分頁內容，使用的是 `slice()` 方法，如下所示：

```
$('#books tr').hide().slice(0, rowsPage).show();
```

上述程式碼選擇所有表格列後，使用 `hide()` 方法隱藏表格列，然後使用 `slice()` 方法取出部分表格列，方法的 2 個參數是索引範圍的開始與結束（從 0 開始，但不包含結束索引），以此例是 `slice(0, 3)`，可以取出 0、1 和 2 共三列，最後使用 `show()` 方法顯示部分表格列。

因為是第 1 頁，所以我們需要替頁碼超連結的第 1 個 a 元素套用 active 樣式類別，如下所示：

```
$('#nav a:first').addClass('active');
```

點選頁碼超連結顯示其他分頁

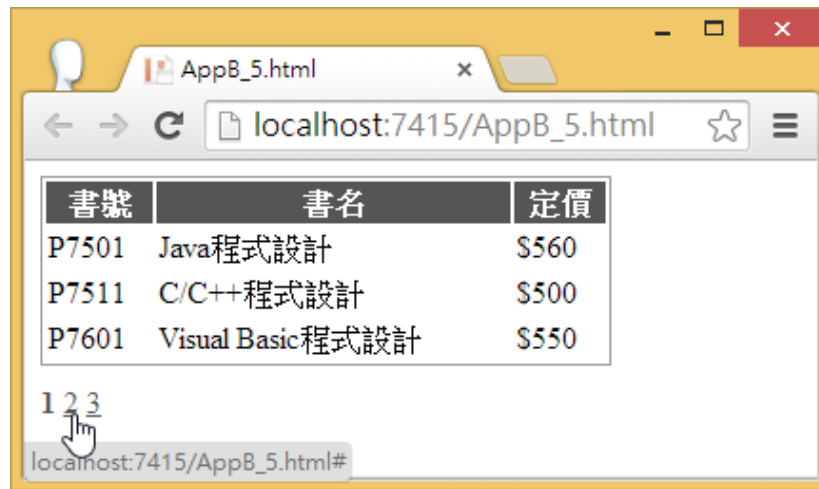
在顯示第 1 頁分頁的表格後，使用者需要按下表格下方的頁碼超連結才能顯示其他分頁，如下所示：

```
$('#nav a').click(function(event) {  
    ...  
    var currPage = $(this).attr('rel');  
    var begin = currPage * rowsPage;  
    var end = begin + rowsPage;  
    $('#books tr').hide().slice(begin, end).show();  
    event.preventDefault();  
});
```

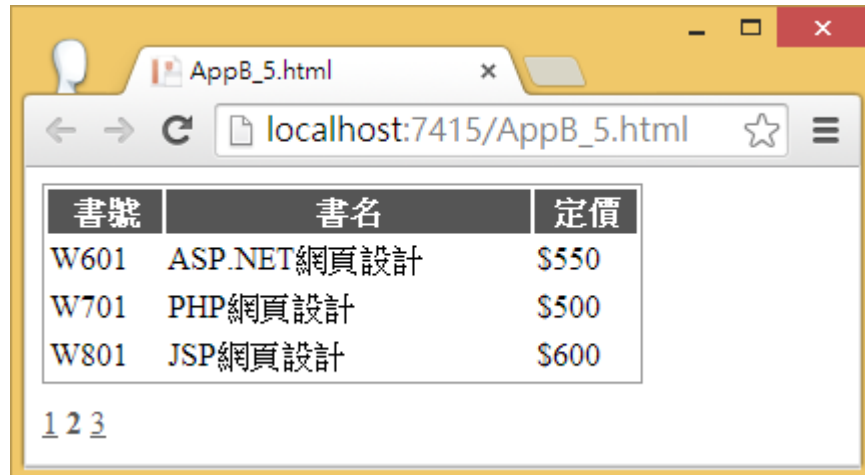
上述程式碼在 a 元素註冊 click 事件，當使用者按下超連結，就取得 rel 屬性的分頁索引，然後計算此分頁的開始和結束索引，呼叫 slice() 方法取出此範圍的表格列來顯示，就可以顯示指定索引頁的分頁表格。

jQuery 程式：AppB_5.html

在 jQuery 程式建立分頁顯示圖書資料的表格，共有 3 頁，首先顯示第 1 頁的 3 列，如下圖所示：



點選頁碼超連結【2】，可以顯示第 2 頁分頁的表格內容，如下圖所示：



程式內容

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
```

```
05: <title>AppB_5.html</title>
06: <style type="text/css">
07: th { background-color : #555; color: white; }
08: table, #books { border : 1px solid #999;
09:                 width : 300px;}
10: #nav { margin : 10px 0;}
11: #nav a { color : #555; }
12: .active { text-decoration : none;
13:          font-weight : bold; }
14: </style>
15: <script src="jquery-2.1.1.min.js"></script>
16: <script>
17: $(document).ready(function(){
18:     var rowsPage = 3; // 每頁有幾列
19:     // 取得表格共幾列
20:     var totalRows = $('#books tr').length;
21:     // 計算頁數
22:     var numPages = Math.floor(totalRows/rowsPage);
23:     var mod = totalRows % rowsPage; // 餘數
24:     if (mod > 0) numPages++; // 不能整除
25:     for ( i = 0; i < numPages;i++ ) {
26:         var pageNum = i + 1; // 頁碼超連結
27:         $('#nav').append('<a href="#" rel=" ' +
28:             i + ">' + pageNum + '</a> ');
29:     } // 顯示第 1 頁
30:     $('#books tr').hide().slice(0, rowsPage).show();
31:     $('#nav a:first').addClass('active');
32:     // 按下頁碼超連結
33:     $('#nav a').click(function(event) {
34:         $('#nav a').removeClass('active');
35:         $(this).addClass('active');
36:         var currPage = $(this).attr('rel');
37:         var begin = currPage * rowsPage;
38:         var end = begin + rowsPage;
39:         $('#books tr').hide().slice(begin, end).show();
40:         event.preventDefault();
41:     });
42: });
43: </script>
44: </head>
45: <body>
46: <div class="content">
47:     <table>
```

```
48:     <thead>
49:         <tr>
50:             <th>書號</th><th>書名</th><th>定價</th>
51:         </tr>
52:     </thead>
53:     <tbody id="books">
54:         <tr>
55:             <td>P7501</td>
56:             <td>Java 程式設計</td>
57:             <td>$560</td>
58:         </tr><tr>
59:             <td>P7511</td>
60:             <td>C/C++ 程式設計</td>
61:             <td>$500</td>
62:         </tr><tr>
63:             <td>P7601</td>
64:             <td>Visual Basic 程式設計</td>
65:             <td>$550</td>
66:         </tr><tr>
67:             <td>W601</td>
68:             <td>ASP.NET 網頁設計</td>
69:             <td>$550</td>
70:         </tr><tr>
71:             <td>W701</td>
72:             <td>PHP 網頁設計</td>
73:             <td>$500</td>
74:         </tr><tr>
75:             <td>W801</td>
76:             <td>JSP 網頁設計</td>
77:             <td>$600</td>
78:         </tr><tr>
79:             <td>W901</td>
80:             <td>JavaScript 網頁設計</td>
81:             <td>$600</td>
82:         </tr>
83:     </tbody>
84: </table>
85: <div id="nav"></div>
86: </div>
87: </body>
88: </html>
```

程式說明

- 第 6~14 列：HTML 網頁套用的 CSS 樣式，這是套用在第 47~85 列 table、th、div 和 a 元素的 CSS 樣式。
- 第 7~9 列：分別套用在第 50 列 th 元素和第 53~83 列表格 tbody 元素的 CSS 樣式。
- 第 10~11 列：套用在第 85 列 div 元素和其中 a 元素的 CSS 樣式。
- 第 12~13 列：目前頁碼超連結套用的 CSS 樣式，不顯示超連結的底線。
- 第 18~20 列：指定每頁有幾列的 rowsPage 變數，在第 20 列使用 length 屬性取得表格共幾列。
- 第 22 列：計算表格的分頁數。
- 第 23~24 列：如果不能整除，使用 if 條件將頁數加一。
- 第 25~29 列：使用 for 迴圈顯示表格下方的頁碼超連結。
- 第 30~31 列：顯示第 1 頁分頁表格，和在第 1 頁的頁碼超連結套用目前分頁的 CSS 樣式。
- 第 33~41 列：頁碼超連結 a 元素的 click 事件處理，匿名函數是在第 34~35 列更改顯示目前分頁樣式的超連結，第 36 列取得 rel 屬性的分頁索引，第 37~38 列重新計算此分頁的表格列範圍，然後在第 39 列顯示此分頁的表格。