

第 1 2 章

CSS 與 JavaScript

12-1 宣告CSS	3
12-2 使用CSS	7
12-2-1 標準用法.....	8
12-2-2 進階用法.....	10
12-2-3 樣式宣告衝突時的優先順序	14
12-3 動態樣式	16
12-3-1 動態變更樣式	17
12-3-2 非標籤內動態變更樣式.....	21
12-4 元件定位	27
12-5 隱藏元件	錯誤！ 尚未定義書籤。

CSS？這是什麼東東啊？！相信你看到這三個字母時必定與偶一樣：丈二金剛摸不著頭腦，其實 CSS 的全名是〔Cascading Style Sheet〕，以字面上來解釋就是〔連(串)接樣式表〕，這種說法不要說是剛接觸的你，連偶都搞不懂！簡單的說，就是網頁的格式設定啦！



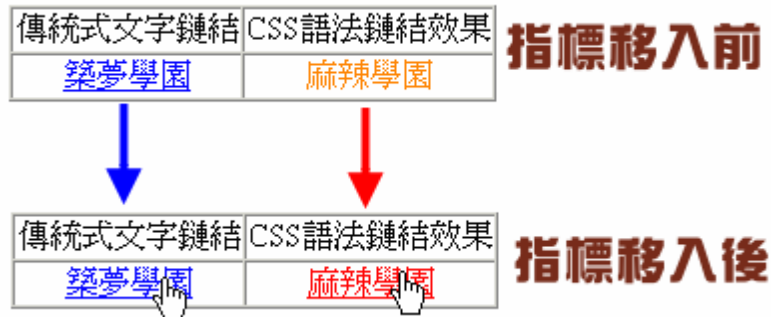
使用 CSS 有什麼好處呢？

➡ 通吃瀏覽器：CSS 樣式表，不管瀏覽者是使用何種瀏覽器來觀賞網頁皆能表現出網頁獨特的風格，不單是市場上使用率最高的 Microsoft Internet Explorer (IE) 與 Netscape Communicator 瀏覽器支援，甚至許多非主流的瀏覽器，如 Emacs-W3、Arena.....等，都有支援解析的能力。

➡ 網路環保：在大家全力推廣全民上網的同時，出現了一個嚴重的危機〔網路頻寬不足〕，一方面是因為上網人口的急速增加造成頻寬不足，但最大的原因是網頁設計者在網頁設計過程中使用了大量的圖片，圖片檔的體積大，在傳輸的過程中消耗了大量的網路資源，而 CSS 的出現將使這個問題獲得部分疏解，CSS 可使用單純的文字來達到與圖片相同的表現效果！也就是說：一段簡短的文字就可以取代體積龐大的圖片！不但傳輸速度加快，同時也獲得與圖片相同的視覺效果。

➡ 特立獨行的風格：一個人氣旺盛的網站之所以吸引人就是因為擁有獨一無二的風格！舉例來說：在 CSS 樣式表出現之前，幾乎所有網頁中的超鏈結文字都會加上底線 (Under

Line) ，而且並無特色變化，如果加上 CSS 語法那就大大不同啦！下方的兩個超鏈結文字樣式，你喜歡哪一個？



👉 **網頁維護容易**：為了維持我們特立獨行的網站風格，同時也為了維持網頁的一致性，我們可以建立一個獨立的 CSS 樣式表檔案，讓我們建立的每一個網頁樣式都能參照此一樣式檔案，當此一樣式檔案內容改變後，所有有參照此一樣式檔案的網頁外觀也就會全部跟著改觀了，因此我們不必一一的修改各個網頁來達到網站風格的一致性，如果你維護的網頁只有幾頁，或許並不會感受到它的便利性，如果是上百頁甚至數百頁，你就會體會到它所帶來的便利性啦！

12-1 宣告 CSS

要宣告 CSS 樣式表首先你必須具備有 HTML 網頁語法的基礎！也就是你必須了解網頁原始碼中標籤的意義，在此我們假設

你已經熟悉了這些 HTML 網頁標籤 (HTML 基本標籤請參考附錄) !



樣式宣告的標準語法如下 :

Element {Property : value}

➡ **Element** :HTML 網頁標籤名稱 , 如 : <P>、</P>.....等。

➡ **Property**:HTML 網頁標籤屬性名稱,如 :color, font-size.....等。

➡ **value**:HTML 網頁標籤屬性值。

注意 : HTML 網頁標籤屬性名稱與 HTML 網頁標籤屬性值之間必須以冒號「:」隔開 ; 而 HTML 網頁標籤屬性名稱與 HTML 網頁標籤屬性值則必需以一組大括號{ }包括起來 !



獨立宣告

獨立宣告 ? 什麼叫獨立宣告啊 ? ! 獨立宣告也就所謂的**分組宣告** , 網頁標籤的可設定屬性相當多 , 因此以分組的宣告方式來管理是一個不錯的選擇 , 例如 : BODY 標籤它可以設定的屬性就相當地多 , 有背景顏色、文字大小、文字字型樣式.....等等 , 現在就以 BODY 標籤的樣式宣告為示範對象進行解說 !

➡ **範例解說 A**，如果我們在樣式表中宣告下列的 BODY 標籤敘述：

```
BODY { background : #FF0000 ; font - size : 10pt }
```

這樣的樣式宣告將會把包括在 < BODY > 與 < /BODY > 之間的文字外觀顏色都變成紅色同時將文字的大小指定為 10pt。

➡ **範例解說 B**，如果我們在樣式表中宣告下列的 BODY 標籤敘述：

```
BODY { background : #00FF00 }  
BODY { font - size : 10pt }
```

這樣的樣式宣告又會造成什麼結果呢？結果是與範例解說 A 一模一樣的：被包括在 < BODY > 與 < /BODY > 之間的文字外觀顏色都變成紅色而且文字的大小指定為 10pt。既然所得到的結果都一樣，那分組的意義在哪裡呢？**分組的意義在於分類管理標籤不同的樣式性質**，以本例來說，我們分類管理了 BODY 標籤的顏色相關屬性與文字相關屬性！



複合宣告

複合宣告，複合宣告可分為兩種情況！一種是將網頁標籤的不同樣式性質宣告在一起〔**不同的屬性設定以分號格開**〕，這種情況之前我們已經使用過了呦！另一種就是將不同的網頁標籤但

是樣式性質相同的屬性合併在一起宣告〔不同的網頁標籤以逗號格開〕！

➡ 單獨標籤的複合宣告：

A 一行通吃

```
BODY{background:#FF0000;color:#00FF00;font-size:10pt;font-family:細明體 }
```

這樣的樣式宣告結果將是：被包括在 `<BODY>` 與 `</BODY>` 標籤之間的文字外觀顏色都變成綠色、文字的大小指定為 10pt、文字字型為「細明體」，同時，背景顏色也被指定為紅色。在本例中我們一次就宣告了 `BODY` 標籤的四種樣式屬性，但請注意：每個指定的樣式屬性之間必須以分號隔開！

B 多行排列

```
BODY { background:#FF0000 ;  
        color:#00FF00 ;  
        font-size:10pt ;  
        font-family:細明體 }
```

其實多行排列與一行通吃所得到的結果是相同的！但是為了顧及 HTML 網頁原始檔的維護與排版，當宣告過長時我們就可在分號之後加以折行撰寫，不管中間有多少空白字元或是幾個換行列，只要我們宣告的格式正確就都不會有錯誤發生的呦！

➡ 單獨標籤的複合宣告：

A 一行通吃

```
BODY , P { color:#00FF00 ; font-size:10pt ; font-family:細明體 }
```

這樣的樣式宣告結果將是：被包括在 `<BODY>` 與 `</BODY>` 標籤之間的文字外觀顏色都變成綠色、文字的大小指定為 10pt、文字字型為細明體；同時，在被包括在 `<P>` 與 `</P>` 標籤之間的文字外觀顏色也都變成綠色、文字的大小指定為 10pt、文字字型為細明體，本例中我們一次就宣告了 `BODY` 標籤與 `P` 標籤的三種共通樣式屬性。

請注意：進行複合宣告的標籤之間必須以逗號隔開！

B 多行排列

```
BODY , P { color:#00FF00 ;  
           font-size:10pt ;  
           font-family:細明體 }
```

同樣的，多組標籤的複合宣告也可以使用多行排列，與一行通吃所得到的結果是相同的！

12-2 使用 CSS

為了讓大家了解 CSS 的使用方式我們將以循序漸漸的方式引導大家進入 CSS 的美麗新世界！

➡ 樣式表宣告的位置：樣式表宣告必須被包括在網頁的 `<HEAD>` 與 `</HEAD>` 之間。

➡ 樣式表宣告的範圍：樣式表宣告的內容必須被包括在 `<STYLE>` 與 `</STYLE>` 標籤之間。

➡ 危險的陷阱：為了避免瀏覽器不支援 CSS 的語法性質，我們必須於 CSS 宣告內容的兩端加上 `<!--` 與 `-->` 的註解符號！若不加上者一組註解符號會產生什麼後果呢？若瀏覽者所使用的瀏覽器並不支援 CSS 的性質屬性，我們所書寫的這些 CSS 宣告將會被當成一般的網頁文字而顯示在網頁上！

12-2-1 標準用法

➡ 典型的 `<STYLE>` 樣式表宣告：

12-2.htm

```
1. <HTML>
2. <HEAD>
3. <STYLE>
4. <!--
5. BODY {background:RED;color:BLACK}
6. BODY {font-size:12pt}
7. -->
8. </STYLE>
```


9. <TITLE>典型的樣式表宣告</TITLE>
10. </HEAD>
11. <BODY>這是標準的 STYLE 宣告</BODY>
12. </HTML>

這樣的樣式宣告會造成什麼結果呢？結果就是將被包括在 <BODY>與<BODY>標籤之間的文字外觀顏色都變成黑色、文字大小為 12pt，同時網頁的背景色設定為紅色。



➡ 網頁標籤內的<STYLE>樣式宣告：

我們除了可以將 CSS 屬性以集合的方式合併宣告外，也可以在網頁標籤中直接加入 CSS 的樣式性質！

12-3.htm

1. <HTML>
2. <HEAD>
3. <TITLE>網頁標籤內的樣式宣告</TITLE>
4. </HEAD>
5. <BODY>

6. `<P STYLE=" font-size:14pt;color:BLUE">`

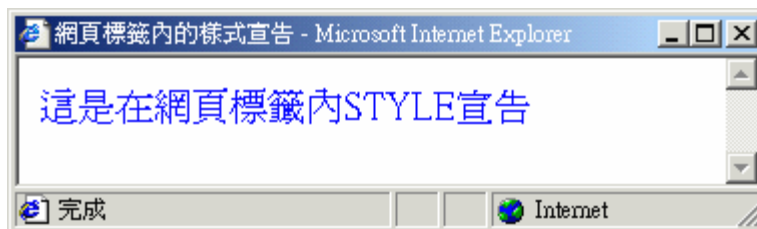
7. 這是在網頁標籤內 STYLE 宣告

8. `</P>`

9. `</BODY>`

10. `</HTML>`

這樣的樣式宣告又將會造成什麼結果呢？結果就是將被包括在 `<P>` 與 `</P>` 標籤之間的文字外觀顏色都變成藍色、文字大小為 14pt，為什麼會這樣呢？因為我們在 `<P>` 網頁標籤中又加上了 CSS 的 `STYLE` 性質設定：`font-size:14pt` 將文字大小格式設定為 14pt；`color:BLUE` 同時將文字外觀顏色設定為藍色。



提示！！

直接以 `STYLE` 屬性在 `HTML` 標籤內指定樣式，就是所謂的『線內樣式』設定。

12-2-2 進階用法

➡ 進階的 `<LINK>` 樣式表檔案參照：

如果我們有很多的檔案要採用統一的设计風格，如果在每張網頁中都加入相同的樣式宣告未免太麻煩、太沒效率啦！因此，我們可以先自行建立一個樣式參照檔案，這個參照檔案只是一個純文字的檔案格式，任意一種文書編輯器就可以用來編輯參照檔案的內容！

樣式表檔案的內容並不需要加上 `<STYLE>` 與 `</STYLE>` 標籤，同時，更不需要加上 `<!--` 與 `-->` 註解標籤，現在就請你開啟你慣用的文書編輯器來編輯一個樣式參照檔案，同時將檔案名稱定為 `test1.css` (這只是範例，參照檔案的名稱你可以自訂)並且加入以下內容：

test1.css

1. `BODY {font-size: 12pt ; font-family: 標楷體}`
2. `BODY {background: #008800 ; color: #FFFFFF}`
- 3.
4. `P {color:RED}`
5. `P {font-size:14pt}`

這樣的樣式內容是什麼意思呢？就是將被包括在 `<BODY>` 與 `</BODY>` 標籤之間的文字字型設定為標楷體、文字大小為 12pt，網頁的背景設定為綠色，文字外觀顏色都變成白色；包括在 `<P>` 與 `</P>` 標籤之間的文字大小為 14pt 文字外觀顏色都變成紅色。

在建立好參照檔案後，接著我們就要利用 **LINK 標籤**來指定樣式表的關聯啦！現在我們就來建立一個 HTML 網頁檔：

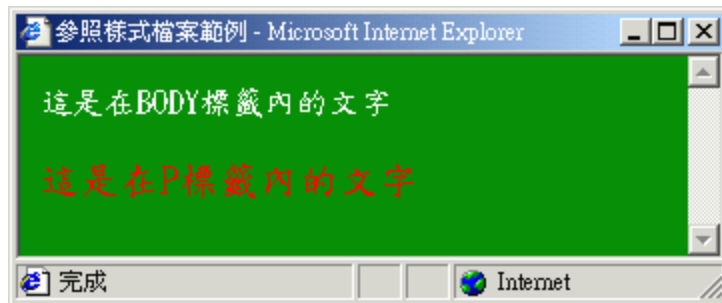
12-4.htm

```
1. <HTML>
2. <HEAD>
3. <TITLE>參照樣式檔案範例</TITLE>
4. <LINK REL=STYLESHEET TYPE="text/css" HREF="test.css">
5. </HEAD>
6. <BODY>
7. 這是在 BODY 標籤內的文字
8. <P>
9. 這是在 P 標籤內的文字
10. </P>
11. </BODY>
12. </HTML>
```

A REL=STYLESHEET：表示我們要參照一個外部的樣式檔案。

B TYPE="text/css"：說明我們參照的檔案類型。

C HREF="test.css"：指定我們要參照的樣式檔案位置及檔名。



➤ 進階的 @import 樣式表檔案參照：

我們除了可以用 [LINK 標籤](#) 來指定樣式表的關聯外，也可以採用 [@import](#) 指定法！但是這種指定法比較麻煩一點，不但要把它加在 [<STYLE>](#) 與 [</STYLE>](#) 標籤內，同時還得加上 [<!--](#) 與 [-->](#) 註解標籤。

我們沿用上一個範例的 [test1.css](#) 樣式檔案來說明，現在就讓我們來建立一個以 [@import](#) 指定法的 HTML 網頁檔：

12-5.htm

1. <HTML>
2. <HEAD>
3. <TITLE>參照樣式檔案範例</TITLE>
4. <STYLE>
5. <!--
6. @import url("test1.css");
7. -->

```
8. </STYLE>
9. </HEAD>
10. <BODY>
11. 這是在 BODY 標籤內的文字
12. <P>
13. 這是在 P 標籤內的文字
14. </P>
15. </BODY>
16. </HTML>
```

A `URL(/.test.css)`;指定我們要參照的樣式檔案位置及檔名。

B 使用 `@import` 指定法來參照外部樣式檔案；請注意：在 `@import` 敘述的最後面一定要加上一個「；」分號！

12-2-3 樣式宣告衝突時的優先順序

當我們混合多種的樣式宣告法後，若有些標籤樣式的性質重複宣告了怎麼辦？他們的優先順序如何？。

A 作用優先順序：直接加註在標籤內的樣式性質 > STYLE 樣式區段內的性質 > 鏈結進來的樣式檔案性質。

- **直接加註在標籤內的樣式性質**：例如，`<P style="color:YELLOW">`。

- **STYLE 樣式區段內的性質**：被包括在 <STYLE> 與 </STYLE> 標籤之間的樣式性質設定。
- **鏈結進來的樣式檔案性質**：利用 LINK 方法或 @import 方法鏈結進來的樣式檔案宣告內容。

B 設定優先順序：網頁設計者的設定 > 瀏覽者的使用者自訂 > 瀏覽器的預先設定。

- **網頁設計者的設定**：就是我們所討論的樣式性質設定啦！
- **瀏覽者的使用者自訂**：瀏覽者透過所使用的瀏覽器進行自己瀏覽格式的設定。
- **瀏覽器的預先設定**：瀏覽者所使用的瀏覽器所預設的瀏覽格式。



提示！！

一般來說，我們都只是運用 **CSS** 來設定網頁文字的大小、樣式、字型與顏色，但是現在最流行的是所謂的：**動態網頁 (Dynamic HTML)**，動態網頁解決了所有對網頁文件的存取限制，也就說我們可以在 **HTML** 網頁中加入動態的效果，動態的 **HTML** 可以使 **HTML** 元件(圖片、文字字串等等)游移於網頁之中，或是產生與瀏覽者有互動性的增強功能，當然，單靠 **CSS** 是做不到這種互動性功能的，但是卻可以結合程式語言來達到動態的效果，而最常用來結合 **CSS** 創造出動態 **HTML** 互動性效果的程式語言應該就是 **JavaScript** 了。

12-3 動態樣式

網頁標籤，如：`<P>`、`<BODY>`...等，都是網頁中的元件，當我們在 CSS 串列表中將網頁元件的屬性加以設定後，則網頁中凡是被該元件標籤包括者，其樣式將全部相同，如下例：

12-6.htm

```
1. <HTML>
2. <HEAD>
3. <TITLE>相同樣式</TITLE>
4. <style type="text/css">
5. <!--
6. p{color:#ff8000;font-size:14pt}
7. //-->
8. </style>
9. </HEAD>
10. <BODY>
11. <P>
12. 這是在 P 標籤內的文字
13. </P>
14. <P>
15. 這也是在 P 標籤內的文字
```



```
16. </P>  
17. </BODY>  
18. </HTML>
```



或許你會說，那我就直接在每個元件標籤中個別以 STYLE 屬性設定，這樣就可以使用相同元件標籤而創造不同樣式的內容了 Y！這是靜態的設定，當然可以（被元件標籤包括者，樣式不變）！如果我要被元件標籤包括者能夠動態改變它的樣式呢？單純的樣式設定是辦不到的！我們必須再加上 JavaScript 的利用才行。

12-3-1 動態變更樣式

假設網頁中有一個超鏈結，我希望在滑鼠指標還沒有碰到它的時候是藍色字、大小 14pt，而當滑鼠碰觸到它的時候，則變成紅色字、大小 16pt，該如何處理？



實例學習：(範例光碟 12-7.htm)

12-7.htm

```
1. <HTML>
2. <HEAD>
3. <TITLE>動態鏈結樣式</TITLE>
4. <style type="text/css">
5. <!--
6. .outlink{color:#0000ff;font-size:14pt}
7. .onlink{color:#ff0000;font-size:16pt}
8. //-->
9. </style>
10. </HEAD>
11. <BODY>
12. <A HREF="http://www.twbts.com"
13. class="outlink"
14. onMouseOver = "this.className='onlink'"
15. onMouseOut = "this.className='outlink'"
16. >
17. 麻辣學園
18. </A>
```

```
19. </BODY>
```

```
20. </HTML>
```

首先我們先利用 CSS 樣式表，建立兩種樣式類別：「outlink」、「onlink」，「outlink」類別為滑鼠指標還沒有碰到超鏈結或碰到超鏈結後又移開的超鏈結外觀樣式，「onlink」類別為滑鼠指標移到超鏈結上超鏈結的外觀樣式。（網頁原始碼粉紅色標註）

在超鏈結<A>標籤中我們設定這個超鏈結的 class 類別屬性為「outlink」，這代表我們預設超鏈結一開始的外觀樣式為 CSS 樣式表「outlink」類別所指定的樣式。當超鏈結標籤發生「onMouseOver」、「onMouseOut」的事件時，再度變更超鏈結所對應的 CSS 樣式表類別所指定的樣式。



提示！！

在 CSS 樣式表中單純的定義一個屬性類別，必須在數行名稱之前加上一個點「.(dot)」，如 12-7.htm 網頁原始碼的第 6、7 行。

動態樣式的變更不只是超鏈結可以使用，表單元件也可以使用。



實例學習：(範例光碟 12-8.htm)

下列這個範例是動態的改變標單按鈕元件的外觀樣式，當滑鼠移至按鈕上時將其背景顏色改變為黑色，文字變成白色，而當滑鼠移開按鈕上時將其背景顏色改變為黃色，文字變成紅色。

12-8.htm

```
1. <HTML>
2. <HEAD>
3. <TITLE>動態按鈕樣式</TITLE>
4. <style type="text/css">
5. <!--
6. .out{color:red;background=yellow}
7. .on{color:white;background=black}
8. //-->
9. </style>
10. </HEAD>
11. <BODY>
12. <FORM>
```

```
13. <INPUT TYPE=BUTTON VALUE="動態按鈕樣式"  
14. class="out"  
15. onMouseOver = "this.className='on'"  
16. onMouseOut = "this.className='out'"  
17. >  
18. </FORM>  
19. </BODY>  
20. </HTML></A>  
21. </BODY>  
22. </HTML>
```



12-3-2 非標籤內動態變更樣式

上一節中，我們都是直接在標籤內設定元件的類別屬性，或是以事件處理器來控制類別屬性的變更，在事件處理器中的程式敘述，我們直接以『this』關鍵字來代替標籤本身的全名，所以我

們並沒有設定標籤的名稱就可以直接變更標籤的屬性，問題是：如果我們的程式碼並不是直接撰寫在事件控制器中，那這改變標籤屬性的程式敘述該如何寫？

要在標籤之外設定其屬性，我們必須為標籤訂立一個名稱屬性，但這個名稱屬性並不是『NAME』！而是『ID』，如下例：

```
<P ID="my">麻辣學園</p>
```

如果我們要改變上例文字的樣式類別，則必須透過它的「ID」屬性來進行：

```
document.all.mylink.className="outlink"
```

我們之前說過：「document 物件指的就是網頁文件本身」，那我們為何在上式中不直接寫成：

```
document.mylink.className="outlink"
```

因為在我們使用 CSS 樣式表來定義網頁元件樣式時，我們的 JavaScript 不可以直接以元件的名稱去控制，而必須透過「all」屬性來控制，「all」就代表網頁文件中全部的元件集合，所以沒有透過 all 屬性就無法控制 HTML 標籤元件。



實例學習：(範例光碟 12-9.htm)

以事件控制器呼叫函式的方式進行超鏈結樣式設定。

12-9.htm

```
1. <HTML>
2. <HEAD>
3. <TITLE>動態鏈結樣式</TITLE>
4. <style type="text/css">
5. <!--
6. .outlink{color:#0000ff;font-size:14pt}
7. .onlink{color:#ff0000;font-size:16pt}
8. //-->
9. </style>
10.
11. <SCRIPT Language="JavaScript">
12. function change(stau)
13. {
14. if (stau=="in")
15. {document.all.mylink.className="onlink"}
16. if (stau=="out")
17. {document.all.mylink.className="outlink"}
18. }
19. </SCRIPT>
20.
21. </HEAD>
```

```
22. <BODY>
23. <A HREF="http://www.twbts.com"
24. ID="mylink"
25. onMouseOver = "change('in')"
26. onMouseOut = "change('out')"
27. >
28. 麻辣學園
29. </A>
30. </BODY>
31. </HTML>
```

- ➡ 在程式碼第 23-27 行中，我們建立了一個超鏈結標籤，並設定其 ID 為「mylink」，當滑鼠指標移入或移出超鏈結時，都會呼叫 change() 函式來動態改變超鏈結的外觀樣式，不過，在滑鼠指標移入及移出超鏈結時，傳遞給 change() 函式的參數內容是不同的，「in」代表滑鼠指標移入超鏈結，「out」為滑鼠指標移出超鏈結。
- ➡ 程式碼第 11-19 行，當 change() 函式被呼叫時，將會進行參數內容的判斷，如果是「in」則將名稱為「mylink」的網頁元件的樣式類別設定為「onlink」；如果是「out」則將名稱為「mylink」的網頁元件的樣式類別設定為「outlink」。



以改變類別設定達到變更網頁元件的外觀是非常方便，但如果只是要改變網頁元件的某一種樣式屬性呢？仍然必須使用類別設定嗎？這倒不必，我們可以直接透過「style」屬性來做就可以了，下表為「style」屬性常見的五種樣式屬性：

屬性	作用
fontSize	字型字體大小
fontWeight	字型樣式 (如：斜體)
fontFamily	字型(如：細明體)
color	顏色
background	背景



實例學習：(範例光碟 12-10.htm)

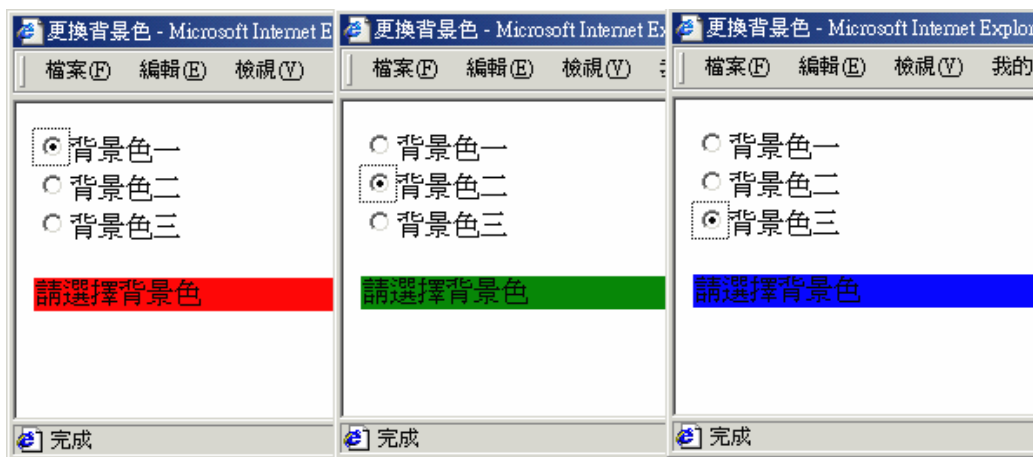
建立一程式，在網頁中佈置表單單選按鈕元件，由瀏覽者自行決定文字字串背景圖。

12-10.htm

1. <HTML>

```
2. <HEAD>
3. <TITLE>更換背景色</TITLE>
4.
5. <SCRIPT Language="JavaScript">
6. function change(color)
7. {
8. document.all.mycolor.style.background=color
9. }
10. </SCRIPT>
11.
12. </HEAD>
13. <BODY>
14. <form>
15. <input type="radio" name="r" onClick="change('red')">背景色一<br>
16. <input type="radio" name="r" onClick="change('green')">背景色二<br>
17. <input type="radio" name="r" onClick="change('blue')">背景色三<br>
18. </form>
19.
20. <DIV ID="mycolor">
21. 請選擇背景色
22. </DIV>
```

- 23.
- 24. </BODY>
- 25. </HTML>



12-4 元件定位