

JavaScript 電子書範例

我們在前面的篇幅，瞭解了如何用 HTML 來呈現資料的邏輯結構，用 CSS 來呈現網頁的視覺外觀，在本章我們用 JavaScript 做出一個整合應用的電子書範例。

第一節 CSS 與版面

請用瀏覽器開啟檔案 `ebook1-1.htm`，我們會看到一張繪本的圖片、一個左箭頭、一個右箭頭、以及一個奇幻世界的美麗公主小圖。這是一個還沒有經過適當的 CSS 排版的頁面，請你練習看看，是否能夠運用前面所學到的 CSS 版面概念，將這些小圖示排到正確的位置。如果你無法使用 CSS 隨心所欲的排列它們，那麼你也可以參考檔案 `ebook1-2.htm` 裡面的 CSS 定義。

第二節 DOM 與 JavaScript 互動

有了簡單的電子書版面之後，我們現在要開始為它加入互動。請打開檔案 `ebook1-3.htm`。左按鈕的容器是以下的標籤

```
<div id="leftArrow" onclick="prevPage();">
```

這樣的標籤有兩個重要的地方，首先它的 id 是 `leftArrow`，一個標籤一定要有 id，後續才可以透過 JavaScript 的程式碼去控制它。其次，我們看到它的 `onclick` 代表了滑鼠點擊時，要去觸發的回應函數，是 `prevPage()`；這個函數。整合在一起來看，也就是說，當滑鼠點擊了向左的箭時，要執行 `prevPage` 函數以翻到上一頁。類似的道理，在右箭頭的容器上

```
<div id="rightArrow" onclick="nextPage();">
```

則是說，當滑鼠點擊了向右的箭時，要執行 `nextPage` 函數以翻到下一頁。要如何讓電腦產生換頁的動作呢？請看到資料夾下有一個子資料夾 `image`，裡面有 `1.jpg`、`2.jpg`、`3.jpg` 等等，代表這本繪本的第一頁、第二頁、和第三頁等等。我們在網頁一打開時，已經載入了 `1.jpg`，這不是透過程式的控制，而是一個我們本來就熟悉的 HTML 頁面

```
<div id="pictureContainer">
  
</div>
```

而這個檔案裡的 JavaScript

```
39 <script type="text/javascript">
40   var currentPage=1;
41   function nextPage()
42   {
43     currentPage++;
44     document.getElementById("picture").src="image/"+currentPage+".jpg";
45   }
```

```

46    function prevPage()
47    {
48        currentPage--;
49        document.getElementById("picture").src="image/"+currentPage+".jpg";
50    }
51    </script>

```

在第 40 行宣告了一個變數 `currentPage`，代表目前所在的頁面是第一頁。當我們執行換頁時，以 `nextPage` 函數來說，它先將 `currentPage++`，也就是把 `currentPage` 這個變數的值，變成它原來的值再加 1，假設我們現在是在第一頁，經過這行指令的作用，`currentPage` 的值就從 1 變成 2。然後第 35 行

`document.getElementById("picture").src="image/"+currentPage+".jpg";`
`document.getElementById("picture")`指的就是 id 為 `picture` 的那個標籤，也就是我們原來載入繪本的第一張照片的地方。`document.getElementById("picture").src` 指的是這個標籤 `` 的 `src` 屬性，這個屬性原來是 `image/1.jpg`，會載入 `image` 資料夾下的 `1.jpg` 那張圖，我們現在將 `document.getElementById("picture").src` 設定為 `"image/"+currentPage+".jpg"`，也就是 `"image/"+2+".jpg"`，也就是 `"image/2.jpg"`，因此它會動態的把原來的圖卸除掉，並且載入新的圖，也就是第二頁。你可以自由的按下一頁、上一頁試試看，它可以正常的換頁，但是碰到超出範圍時(第 20 頁、第 0 頁)，就會變成無法顯示的狀態。

檔案 `ebook1-4.htm` 修正了超出頁面範圍的問題，它設置了變數 `totalPage`，

```

41    var totalPages=19;

```

並且在換頁前先進行判斷

```

42    function nextPage()
43    {
44        if (currentPage<totalPages)
45        {
46            currentPage++;
47            document.getElementById("picture").src="image/"+currentPage+".jpg";
48        }
49    }

```

這樣就可以有效的確認，只會翻到合法的頁面，而不會跑到不正確的範圍。再進一步，我們發現了這樣的運作雖然正確，但是瀏覽網站的人，當它們發現到按下右鍵並沒有反應時，並不知道是因為所有的頁面都已經結束，還是因為網路傳輸

速度太慢，又或者是他按得不夠用力。一個好的介面設計應該要避免這樣的誤會發生，所以當沒有下一頁時，我們應該直接將下一頁的按鈕隱藏起來。檔案 ebook1-5.htm 增加了一個 hideArrow 函數

```
function hideArrow()
{
    if (currentPage==1)
    {
        document.getElementById("leftArrow").style.visibility="hidden";
    }
    else
    {
        document.getElementById("leftArrow").style.visibility="visible";
    }
    if (currentPage==totalPages)
    {
        document.getElementById("rightArrow").style.visibility="hidden";
    }
    else
    {
        document.getElementById("rightArrow").style.visibility="visible";
    }
}
```

這個函數會判斷我們現在處於哪一個頁面，如果已經沒有下一頁，就不要顯示下一頁的按鈕，同樣的，如果已經沒有上一頁，就不要顯示上一頁的按鈕。在原來的換頁函數 nextPage 和 prevPage 中，最後都要呼叫 hideArrow 函數以重新確認按鈕的狀態。這裡隱藏按鈕的方式是

```
document.getElementById("leftArrow").style.visibility="hidden";
```

它等於是透過 JavaScript，動態的改變一個標籤的 CSS 屬性值。整個換頁功能看起來都正常了，但是一開始的時候，上一頁並沒有隱藏，因為我們是透過換頁時才檢查，一開始並沒有執行到換頁的函數。所以最後我們在檔案 ebook1-6.htm 中，藉由<body>標籤的事件

```
<body onLoad="hideArrow();">
```

當這個頁面一被完整載入時，就會馬上執行一次 hideArrow 函數。從這些例子我們看到幾個觀念：

- (1) 標籤透過識別 ID 讓 JavaScript 取得控制，
- (2) JavaScript 可以在網頁的執行過程中，動態的改變標籤的屬性，以改變網頁的資料內容，
- (3) JavaScript 可以在網頁的執行過程中，動態的改變標籤的 CSS 屬性，以改變網

頁的視覺外觀，

(4)透過事件偵聽與回應的模式，可以達到互動功能。

第三節 HTML 5 與 JavaScript 聲音控制

有了繪本的畫面以後，我們現在開始要加入一些好聽的背景音樂。檔案 ebook2-1.htm 使用的 HTML 5 的聲音播放功能。在原始碼的第 106 行

```
<div><audio id="bgmusic" src="music/p1.mp3" autoplay="true"
loop></audio></div>
```

這裡有一個看不見的聲音控制標籤，它一開始就載入 music 資料夾下的 p1.mp3，設定為自動播放，並且是循環式的。在換頁函數裡，我們都讓它們呼叫了一個新定義的播放音樂的函數

```
81 function loadMusic()
82 {
83     document.getElementById("bgmusic").pause();
84     document.getElementById("bgmusic").src="music/p"+currentPage+".mp3";
85     document.getElementById("bgmusic").play();
86 }
```

你可以試試看，翻到不同的頁面就會有不同的背景音樂。由於 HTML 5 還有各種不同瀏覽器的支援問題，所以如果你用 Chrome 可以聽到正確的效果，但用 IE 就不行。我們可以針對不同的瀏覽器開發程式，使它在網路上有最好的能見度，但是會使程式碼變得複雜許多。由於這只是一個教學的示範例子，就省略這些部份。檔案 ebook2-2.htm 將內部的程式碼重整，它的程式碼變得非常結構化，像是

```
function nextPage()
{
    if (currentPage<totalPages)
    {
        currentPage++;
        loadImage();
        loadMusic();
        hideArrow();
    }
}
```

這樣的形式。每一個函數都各別處理它各自的任務。最後我們在檔案 ebook2-3.htm 把語音也加入了。按下可愛的小公主圖像，就會讀出繪本的泰雅語。在朗讀的過程中，音樂會被設得比較小聲

95 `document.getElementById("bgmusic").volume=0.3;`

朗讀完畢之後才又調回正常音量

```
function readComplete()
{
    document.getElementById("bgmusic").volume=1;
}
```

第四節 動畫效果

請打開檔案 **ebook3-1.htm**，這是一個完全沒有用 **CSS** 排版的 **HTML** 資料。請你把它排成檔案 **ebook3-2.htm** 的樣子，並請注意到圖片上的替代文字。下一個檔案 **ebook3-3.htm** 是基本的 **JavaScript** 互動模式，你可以當作一個簡單的練習。檔案 **ebook3-4.htm** 則示範了一個不一樣的進階技巧。