1 概觀

1.1 網路與網際網路(Network & Internet)

將兩台以上獨立的計算機系統互相連接，使其能進行通訊，即為一個「電腦網路(Computer Network)」，簡稱「網路(Network)」。注意「網路」一詞不一定是指「電腦網路」，還可以指「電路(Electrical Network)」、「社交網路(Social Network)」……等等。電腦網路依照涵蓋範圍由小到大又可細分為：個人網路(Personal Area Network，PAN)、區域網路(Local Area Network，LAN)、城域網路(Metropolitan Area Network，MAN)、廣域網路(Wild Area Network，WAN)，前者通常包含於後者之中。

「網際網路(Internet)」亦稱為「互聯網」，是一種「公共型的廣域網路」。注意「網際網路(Internet)」並不等同於「全球資訊網(World Wide Web，WWW、W3)」，全球資訊網是一項網際網路所提供的服務。除了全球資訊網，網際網路也提供了其他服務，例如：Telnet(電子佈告欄系統，Bulletin Board System、BBS 便是以 Telnet 為基礎建立的)、FTP、E- mail……等等。現今的網際網路使用一種通用的通訊協定，稱為「TCP/IP 協定」，這個協定包含了傳輸控制協議(Transmission Control Protocol，TCP)，以及網際網路協議(Internet Protocol，IP)。與網際網路相對的便是「網內網路(Intranet)」，網內網路指的是一種非公共型的網路，常由區域網路組成，目前普遍地被各企業、組織使用，和網際網路相同，網內網路也使用 TCP/IP 協定。

1.2 全球資訊網與全球資訊網聯盟(W3 & W3C)

全球資訊網是一個由多個互相連結的超文件(Hypertext)所組成的系統，這些超文件仰賴網際網路進行連結。所謂的超文件，是一種非線性文件，使用者可以透過超連結跳躍式地流覽資料，相較之下，一般的線性文件只能一頁一頁循序流覽。目前全球資訊網通用的超文件語言為「超文件標示語言，Hypertext Markup Language、HTML」。

一個完整的全球資訊網包含：伺服器端、用戶端(流覽器，Browser)、超文件、網際網路。全球資訊網聯盟(World Wide Web Consortium，W3C)是一個專門制定全球資訊網的各項標準的組織，包含 HTML、CSS 等基本的網頁語言皆是由全球資訊網聯盟負責制定、更新。

1.3 超文本傳輸協定與網頁伺服器(HTTP & Server)

超文本傳輸協定(Hypertext Transfer Protocol，HTTP)是全球資訊網使用的一項通訊協定，定義了用戶端向伺服器端請求訊息和伺服器端答覆用戶端的標準。為了提升 HTTP 的安全性，另有超文本傳輸安全協議(Hypertext Transfer Protocol Secure，HTTPS)，常用於交易資訊或企業資料等隱密訊息的傳輸。

承上文，網頁伺服器的主要功能為根據用戶端的要求提供網頁，用於網頁的伺服器必

須遵守 HTTP。若網頁內容有使用伺服器端語言，則必須使用支援該伺服器端語言的網頁伺服器才能正常運作。常見的「404 Not Found」，便是網頁伺服器的答覆訊息之一。除了網頁伺服器，其他常見的伺服器還有 FTP Server、Telnet Server……等等。

1.4 伺服器端語言(Server-side Scripting)與用戶端語言(Client-side Scripting)

伺服器端語言(Server-side Scripting)是指在先在伺服器直譯，再將資料傳給用戶端的網頁語言，相對的，用戶端語言(Client-side Scripting)則是指直接將資料傳給用戶端，交由用戶端直譯的網頁語言。由於 HTML 與 CSS 只能單純地呈現內容，若想在網頁上加入互動效果或動態內容，就必須使用其他網頁語言。常見的伺服器端語言有：PHP、ASP、JavaServer Pages (JSP)、Python、Ruby……等等，常見的用戶端語言有：JavaScript、ActionScript、VBScript……等等。

上述的網頁語言皆為腳本語言(Script Language)，這種語言多為直譯式，不需事先編譯成機器語言的執行檔，只要有直譯器(即瀏覽器)就可直接一行一行地執行下去。不過相對的，執行效率就會比較差，在大型程式上尤其明顯。

許多用戶端語言能完成的程式都可用伺服器端語言達成。不過，用戶端語言與伺服器端語言仍有以下特色：一、使用伺服器端語言的網頁需安裝支援該伺服器端語言的網頁伺服器才能開啟，開啟之後伺服器端語言的執行結果會以用戶端語言的形式呈現，而使用用戶端語言的網頁只要有用戶端(即瀏覽器)就可開啟。二、適當使用用戶端語言代替伺服器端語言能減少伺服器的負荷，但不宜將隱密資料用用戶端語言傳遞，以免遭到竊取。一個網頁通常會結合多種用戶端語言與伺服器端語言，而不是只使用一、兩種語言。

1.5 一般編輯器與所見即所得(WYSIWYG)編輯器

撰寫網頁程式碼最直接的方法當然就是使用 Windows 內建的記事本(Notepad)，不過在記事本中所有的程式碼都採用相同樣式，不利於閱讀，因此我們通常會使用有語法高亮度顯示(Syntax Highlighting)功能的編輯器，例如：Notepad++、PSPad。這些編輯器多為免費軟體，能將不同的程式碼用不同的顏色和字體顯示，以便閱讀。

所見即所得(What You See Is What You Get，WYSIWYG)的網頁編輯器能即時顯示程式碼在瀏覽器中呈現的模樣，並准許使用者直接透過圖形化介面(Graphical User Interface，GUI)編輯網頁，而不需親自撰寫程式碼。這些軟體多為商業軟體，可大致分為兩類：一、專業型編輯器，例如：Adobe Dreamweaver、Microsoft Expression Web(較晚發行，普及率不及前者)。二、簡易型編輯器，例如：Microsoft Frontpage(已被 Microsoft Expression Web 取代)、Namo Webeditor。

2 HTML 程式架構

2.1 XML、HTML、XHTML、DHTML、HTML DOM

可擴展標記語言(Extensible Markup Language，XML)是一種邏輯性的文件架構，而不是一種特定的程式語言。將資料以 XML 的形式儲存，可以使其能被電腦讀取，並加以執行。以下即為一個 XML 文件的範例，要讓電腦能讀懂這些內容，必須先定義每一個標籤所代表的意義。

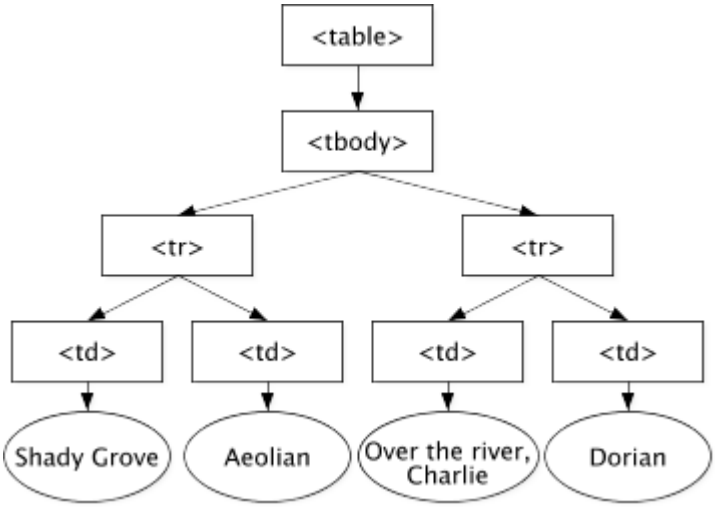
```
<?xml version="1.0" encoding="UTF-8" ?>

<mail>
    <to>Steve Jobs</to>
    <from>Bill Gates</from>
    <title>Greeting</title>
    <content>Hello</content>
</mail>
```

超文件標示語言(Hypertext Markup Language，HTML)是最基本的網頁語言，副檔名使用.html 或.htm 皆可，最新版本為 HTML4.01(HTML5 仍為草案)。HTML 在結構上類似 XML，但較 XML 鬆散，有利不熟悉網頁語言的人使用。多數瀏覽器亦接受了這個現實，能正常顯示語法不夠嚴謹的網頁，但使用不嚴謹的語法，容易產生一些超乎預期的錯誤，故逐漸演化出結構嚴謹、符合 XML 架構的 XHTML。XHTML 的最新版本為 1.1，且與 HTML 相同，皆由 W3C 負責管理。以下為一些 XML 與 XHTML 的比較：

巢狀結構	CORRECT	here is an emphasized paragraph
	INCORRECT	here is an emphasized paragraph
小寫標籤	CORRECT	
	INCORRECT	<LI>
結束標籤	CORRECT	here is a paragraph here is another paragraph
	INCORRECT	here is a paragraph here is another paragraph
完整屬性	CORRECT	<dl compact="compact">
	INCORRECT	<dl compact>
空元素	CORRECT	
	INCORRECT	 <hr>
屬性值	CORRECT	<td rowspan="3">
	INCORRECT	<td rowspan=3>

動態 HTML(Dynamic HTML，DHTML)指的是一種結合 HTML、CSS、用戶端語言、文件物件模型(Document Object Model，DOM)來建立互動性網頁的方法。至於 DOM 指的是將物件的概念融入 HTML 或 XML 中，使開發者可以藉由程式直接控制每一個物件(元素)。專門針

對 HTML 文件的 DOM 稱為 HTML DOM，例如：

程式碼	節點樹
<table> <tbody> <tr> <td>Shady Grove</td> <td>Aeolian</td> </tr> <tr> <td>Over the River, Charlie</td> <td>Dorian</td> </tr> </tbody> </table>	

隨堂練習：試畫出上一頁 XML 文件的 DOM 節點樹。

2.2 HTML 與瀏覽器

由於 HTML 是在用戶端(瀏覽器)直譯，不同的直譯器(瀏覽器)呈現出來的樣貌就會有些許差異。因此，開發者開發時需考量程式在不同瀏覽器間的相容性。

2.3 HTML 文件的結構

一個完整的 HTML 文件包含三個部份：宣告(DOCTYPE)、表頭(HEAD)、內容(BODY)。雖然只要有內容就可以被大多數的瀏覽器正常執行，但仍然建議大家使用比較嚴謹的語法。以下就讓我們來認識這三個重要的部份。

```
<!DOCTYPE html>
<html>

    <head>
        some html code...
    </head>
```

```
    <body>
        some html code...
    </body>

</html>
```

2.4 DOCTYPE 與 DTD

DOCTYPE 是 Document Type 的縮寫，用來宣告你使用的 HTML 規格。HTML4.01 有三種規格，而目前規畫中的 HTML5 只有一種宣告方式，詳見下表：

HTML4.01 Transitional	要求寬鬆， 禁用框架。	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
HTML4.01 Frameset	要求寬鬆， 可用框架。	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
HTML4.01 Strict	要求嚴格， 禁用框架。	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
HTML5	針對 HTML5。	<!DOCTYPE html>

文檔類型定義(Document Type Definition，DTD)定義了 XML 或 HTML 文件中每個元素的意義，以上四種分別使用了不同的 DTD。例如「http://www.w3.org/TR/html4/loose.dtd」即為 HTML4.01 Transitional 的 DTD 檔案的連結路徑。

2.5 HEAD

是 HTML 文件的表頭，可以放入文件的相關訊息，如網頁標題<title>example</title>，或是需要引入的外部檔案，常見的有 CSS、JavaScript 等等，如<script src="example.js" type="text/JavaScript"></script>。

2.6 BODY

這裡放的才是真正要讓網頁顯示出來的內容，包含文字、圖片、影片、表格、表單……等等。

2.7 HTML 標籤的兩種結構

這是大部份 HTML 標籤的結構，但也有少數 HTML 標籤例外，那就是「空元素」的標籤。空元素是一種沒有終止標籤的元素，因為沒有終止標籤，故也不會有標籤之間的內容。以下是一個空元素的範例：

```
<hr width="800" align="center"/>
```

2.8 元素(Element)

元素就是指作用於對象的效果，不同的元素會對標籤之間的內容作不同的效果，例如上一頁的<p>表示一個獨立的段落。空元素沒有作用對象，例如<hr>表示的是繪製出一條分隔線，不需要有標籤之間的內容。

2.9 屬性與屬性值(Attribute & Attribute Value)

添加於起始標籤內，補充說明作用於對象的額外效果，例如<hr width="800">表示這條分隔線的寬度為 800(像素)。在 HTML 中屬性值若為數字，可省略雙引號(")或單引號(')，但在 XHTML 中則不可省略。為避免發生錯誤，建議一律為屬性值加上雙引號或單引號。

2.10 區塊元素與線內元素

「區塊元素」就如同一個容器，在元素標籤中還可再加入其他的元素標籤，相對的，「線內元素」則是一個獨立作用的元素，不可在其中加入別的元素標籤。例如：

```
<form name="example">
    <p>NAME:</p>
    <input type="text" name="name" />
    <p>AGE:</p>
    <input type="text" name="age" />
</form>
```

2.11 加入註解

註解是給開發人員看的，瀏覽器執行時會自動忽略。範例：

```
<!-- This is a comment -->
```

2.12 編碼與特殊符號

一個網頁常會同時使用多種語言，故近幾年來，網頁開發人員多以支援多國語言的萬國碼(Unicode)取代只支援繁體中文的大五碼(Big5)，作為採用的編碼。而在 Unicode 的多種轉換格式(Unicode Transformation Format，UTF)中，又以 UTF-8(8-bit Unicode Transformation Format)最常用於網頁設計。另外，當要在網頁中顯示特殊符號時，為避免與程式碼所使用的符號混淆，多會使用「文字參照」。詳見下表：

特殊符號	名稱實體參照	數值實體參照
"	"	"
&	&	&
<	<	<
>	>	>
§	§	§
«	«	«
»	»	»
©	©	©
®	®	®
™	™	™
±	±	±
¿	¿	¿
¡	¡	¡

2.13 相對路徑與絕對路徑

類型	範例	說明
絕對路徑	http://summer2012.co.cc	永久固定的檔案路徑
相對路徑	example.htm	同一個目錄下的 example.htm 檔案
	folder/example.htm	同一個目錄下 folder 子目錄的 example.htm 檔案
	../ example.htm	上一層目錄下的 example.htm 檔案
	../folder/example.htm	上一層目錄下 folder 子目錄的 example.htm 檔案

使用相對路徑有利於網站資料夾轉移，故較常被使用。

2.14 通用屬性與標準事件

通用屬性是指幾乎所有元素都通用的屬性，標準事件則是用來設定觸發一特定事件的

條件。

類型	項目	說明
通用屬性	id	設定元素的識別名稱，不可重覆，用於篩選元素
	class	設定元素要套用的樣式類別，搭配 CSS 類別選擇器使用
	title	元素的說明文字，滑鼠移至時會顯示
	style	直接輸入 CSS 設定元素要套用的樣式
標準事件	onclick	直接輸入 JavaScript 設定元素被按下時觸發的事件
	onkeydown	直接輸入 JavaScript 設定按下鍵盤時觸發的事件
	onfocus	直接輸入 JavaScript 設定元素被操作時觸發的事件
	onload	直接輸入 JavaScript 設定頁面載入時觸發的事件

2.15 使用 HTML 架構加入其他語言

用戶端語言	CSS	<style type="text/css">... some css code ...</style>
	JavaScript	<script type="text/javascript">... some JavaScript code ...</script>
伺服器端語言	PHP	<?php ... some php code ?>
	ASP	<% ... some asp code %>

3 HTML 語法

3.1 基本元素

元素名稱	說明	屬性或範例
title	設定頁面標題，放在 head 中	<title>example</title>
meta	附加額外資訊，放在 head 中	<meta http-equiv="content-type" content="text/html; charset=utf-8" /> <!-- 以 HTTP 表頭的形式 傳送資訊值給用戶端 -->
h1 ~ h6	放大文字做為標題	!align：對齊方向，包含 left、right、center
div	用於分割文件，區塊元素	!align：對齊方向，包含 left、right、center
span	選取元素內容，線內元素	example
script	直接輸入腳本語言程式碼或引入腳本文件	type：設定使用的 MIME 類型，MIME 定義了多種內容類型(Content-type)，格式為 type / subtype src：設定外部腳本文件的位置
style	直接輸入 CSS 程式碼，放在 head 中	type：設定使用的 MIME 類型，CSS 文件的 MIME 值為 text/css

link	引入 CSS 文件，放在 head 中	type：設定使用的 MIME 類型 href：設定外部腳本文件的位置 rel：設定連結類型(例：rel="stylesheet")

註：[!]符號表示該屬性不可用於 HTML5，應用 CSS 替代。

3.2 文字元素

元素名稱	說明	屬性或範例
p	段落，區塊元素	[!]align：對齊方向，包含 left、right、center
br	斷行，行間無間隔	... ...
a	超連結	href：設定外部腳本文件的位置 target：開啟方式，包含_blank、_self 等等
b	粗體	example
i	斜體	<iexample</i>
sup / sub	上下標	a²+2ab+b²
blockquote	引用區塊，會自動縮排	<blockquote>example</blockquote>
hr	分隔線	[!]align：對齊方向，包含 left、right、center [!]size：寬度，以像素為單位 [!]width：長度，可用像素值或百分比
ul	無編號清單，搭配 li 使用	[!]type：項目符號，包含 disc、circle、square
ol	有編號清單，搭配 li 使用	[!]type：項目符號，包含 1、a、A、i、I
li	清單項目	<ol type="I"> - first - second

註：超連結除了可以使用相對路徑或絕對路徑外，還可使用電子郵件地址或錨點。 電子郵件地址使用範例：text 錨點使用範例：text

3.3 嵌入元素

元素名稱	說明	屬性或範例
img	圖片	src：設定圖片的位置 alt：當圖片無法顯示時的替代文字 width：寬度，可用像素值或百分比 height：高度，可用像素值或百分比 usemap：影像地圖名稱，對應到 map 元素的 id 屬性值 (例：usemap="#mapname") [!]border：框線寬度，以像素為單位
map & area	影像地圖	 <map id="Map"> <area shape="rect" coords="10,10,60,60" href="example.htm" target="_blank " />

		</map> <!-- rect：左上 X,左上 Y,右下 X,右下 Y circle：圓心 X,圓心 Y,半徑 poly：各頂點的 X,Y -->
object & param	物件	<object type="application/x-shockwave-flash" data= "example.swf" width="700" height="500"> <param name="movie" value=" example.swf" /> <param name="quality" value="high" /> </object> <!-- param 元素是用來傳遞參數 給執行該物件的程式 -->
iframe	線內框架	src：設定框架內文件的位置 height：高度，可用像素值或百分比 width：寬度，可用像素值或百分比 □scrolling：捲動軸，值為 auto、yes、no □frameborder：框線，值為 1(顯示)、0(不顯示)

3.4 表格元素

元素名稱	說明	屬性或範例
table	表格	□align：對齊方向，包含 left、right、center □width：寬度，可用像素值或百分比 □cellpadding：儲存格與儲存格內容的間距，可用像素值或百分比 □cellspacing：儲存格間的間距，可用像素值或百分比 □border：框線寬度，以像素為單位
tr	列	□align：對齊方向，包含 left、right、center
td	儲存格	□align：對齊方向，包含 left、right、center □height：高度，可用像素值或百分比 □width：寬度，可用像素值或百分比 □rowspan：合併列 □colspan：合併欄

範例：

```
<table width="100%">
  <tr>
    <td>1,1</td>
    <td>1,2</td>
    <td>1,3</td>
    <td>1,4</td>
  </tr>
```

```
  <tr>
    <td>2,1</td>
    <td rowspan="3">2,2</td>
    <td>2,3</td>
    <td>2,4</td>
  </tr>
  <tr>
    <td>3,1</td>
    <td colspan="2">3,3</td>
  </tr>
  <tr>
    <td>4,1</td>
    <td>4,3</td>
    <td>4,4</td>
  </tr>
</table>
```

隨堂練習：試做出以下表格。

1,1	1,2	
2,1	2,2	2,3
	3,2	3,3

3.5 表單元素

元素名稱	說明	屬性或範例
form	表單	action：表單傳送的目標位置 method：傳送方法，分為 get、post 等等 enctype：資料型態，當表單中含有 intput type ="file"時，必須設定為 multipart/form-data
input	欄位	type：類型，包含 submit、reset、button、hidden、text、raido、checkbox、file 等等 value：設定資料內容 size：寬度，以字數或像素為單位 maxlength：最大字數，用於文字型欄位 checked：預設勾選，用於選擇型欄位 multile：選取多個檔案，用於檔案型欄位
textarea	多行文字欄位	rows：高度，以行為單位 cols：寬度，以字數或像素為單位
select & option	選項清單	<select> <option value="1">1</option> <option value="2" selected="selected">2</option>

		</select>

隨堂練習：試做出一個活動報名表單。

4 CSS 程式架構

4.1 用 CSS 設定樣式

CSS 是一種設定樣式網頁樣式的程式語言，目前最新版本為 CSS2.1(CSS3 仍為草案)。用 CSS 設定樣式有兩種方試，一種是直接使用通用屬性 style，如以下範例 A，另一種則是使用 style 元素搭配 CSS 選擇器，如以下範例 B。從以下兩個例子可以明顯看出，範例 B 的寫法較精簡且可讀性亦較高。

A:

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <table width="500" border="1"
        cellpadding="0" cellspacing="0">
            <tr>
                <td style="font-family:arial;color:red;
                font-size:20px;">example</td>
                <td style="font-family:arial;color:red;
                font-size:20px;">example</td>
                <td style="font-family:arial;color:red;
                font-size:20px;">example</td>
            </tr>
            <tr>
                <td style="font-family:arial;color:red;
                font-size:20px;">example</td>
                <td style="font-family:arial;color:red;
                font-size:20px;">example</td>
                <td style="font-family:arial;color:red;
                font-size:20px;">example</td>
            </tr>
            <tr>
                <td style="font-family:arial;color:red;
                font-size:20px;">example</td>
                <td style="font-family:arial;color:red;
                font-size:20px;">example</td>
```

```
            <td style="font-family:arial;color:red;
            font-size:20px;">example</td>
        </tr>
    </table>
</body>
</html>
```

B:

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
            td {font-family:arial;color:red;font-size:20px;}
        </style>
    </head>
    <body>
        <table width="500" border="1"
        cellpadding="0" cellspacing="0">
            <tr>
                <td>example</td>
                <td>example</td>
                <td>example</td>
            </tr>
            <tr>
                <td>example</td>
                <td>example</td>
                <td>example</td>
            </tr>
            <tr>
                <td>example</td>
                <td>example</td>
                <td>example</td>
            </tr>
        </table>
    </body>
</html>
```

4.2 內部宣告與外部引入

以上示範的兩種方法皆為在內部宣告，若要從外部引入，有兩種方法：

A:

```
<link href="example.css" rel="stylesheet" type="text/css" />
```

B: <style type="text/css"> @import url(example.css); </style>

4.3 CSS 的結構

撰寫 CSS 程式時，有三點事項需特別留意：一、每一組屬性設定的結尾需加上「;」號，但最後一組屬性設定結尾的「;」號可省略。二、屬性值不需以「"」包住，與 HTML 不同。三、若有多個欲套用的對象，需用「,」號隔開。

4.4 加入註解

CSS 中的註解與 C 語言相同，皆是以「/*」開始，「*/」結束。

4.5 標籤選擇器、類別選擇器、ID 選擇器

藉由選擇器可以選擇該樣式要套用的對象，常見的選擇器有以下三種：

類別	範例	說明
標籤選擇器	p { color: red; }	將所有 p 元素套用該樣式
類別選擇器	.eg { color: red; }	將所有 class 屬性之值為 eg 的元素套用該樣式
ID 選擇器	#eg { color: red; }	將所有 id 屬性之值為 eg 的元素套用該樣式

4.6 進階的標籤選擇器

事實上，CSS 的選擇器從以上三種又可延伸出三十幾種，不過多數都不太常被使用，以下就補充介紹三種實用的、設定超連結的「虛擬類別選擇器」：

類別	範例
針對滑鼠從未點過的超連結	a:link { color: red; }
針對滑鼠已經點過的超連結	a:visited { color: red; }
針對滑鼠移至其上的超連結	a:hover { color: red; }

4.7 多重 ID 與標籤複合的選擇器

上述的三種選擇器可以相互混搭使用，例如：

```
#main p .right_text { text-align:right; }
/*
    表示將
    所有 id 屬性之值為 mian 的元素中
    所有 p 元素中
    所有 class 屬性之值為 right_text 的元素
    套用該樣式
*/
```

4.8 CSS 的屬性值

長度	p { font-size: 17 em; }	相對單位，指的是相對於父元素的字體高度 若父元素未設定字體高度，也就是採用瀏覽器的預設值 因多數瀏覽器預設的字體高度皆為 16px 故 1em=16px，0.75em =12px
	p { font-size: 17 ex; }	相對單位，指的是相對於父元素小寫字元的字體高度
	p { font-size: 17 px; }	相對單位，像素
	p { font-size: 17 cm; }	絕對單位，公分
	p { font-size: 17 mm; }	絕對單位，毫米
	p { font-size: 17 pt; }	絕對單位，點(point)
百分比	table { width: 170%; }	讓作用效果隨畫面大小而改變
連結	body { background-image: url(example.png); }	格式為 url(相對或絕對位置)
顏色	p { color: red; }	直接輸入顏色名稱
	p { color: #FF6600; }	設定其 RGB 值，格式為#RRGGBB，三值皆為 16 進制
	p { color: #F60; }	若 R、G、B 值的兩位數字皆同，可省略為#RGB

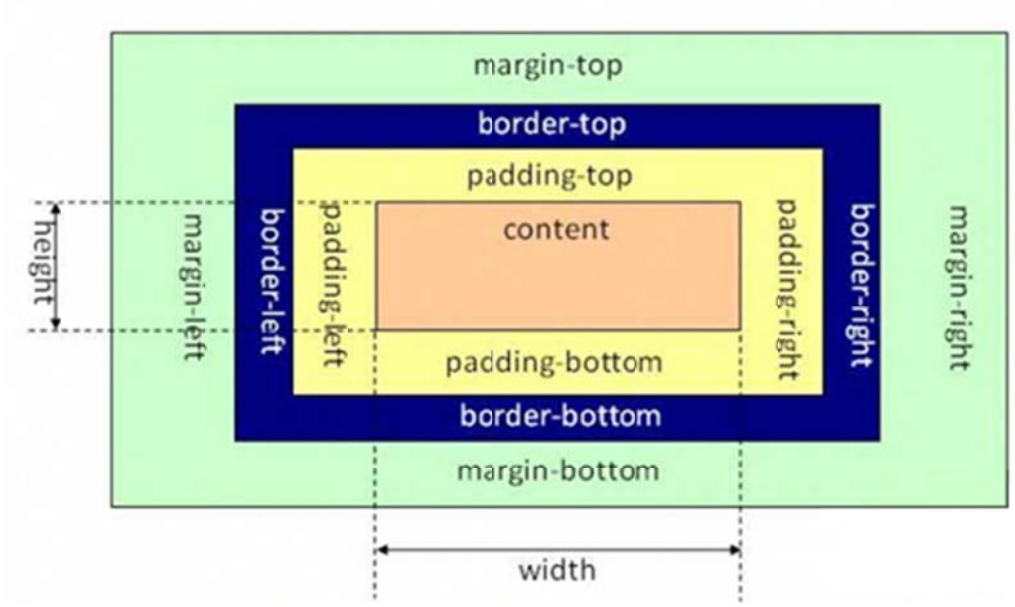
4.9 屬性繼承

在 CSS 中子元素會繼承父元素的樣式，例如以下程式雖然沒有設定 body 中 p 元素的文字顏色，但因 p 是 body 的子元素(包含在 body 之中)，故 p 仍會繼承 body 的樣式設定，而呈現出橘色(#F60)文字。

```
<head>
    <style>
        body { color: #F60; }
    </style>
</head>
<body>
    <p>example</p>
</body>
```

4.10 CSS 元素模型

CSS 將所有元素皆視為一個區塊，由內容(content)、外邊距(margin)、內邊距(padding)、邊框(border)所組成，如下圖所示：




```
<head>
    <style>
        i {
            background-color: #F60;
            padding: 10px;
            border: 5px #69F solid;
            margin: 20px;
        }
    </style>
</head>
<body>
    <p>Example</p>
    <br/>
```

```
    <i>Example</i>
</body>
```

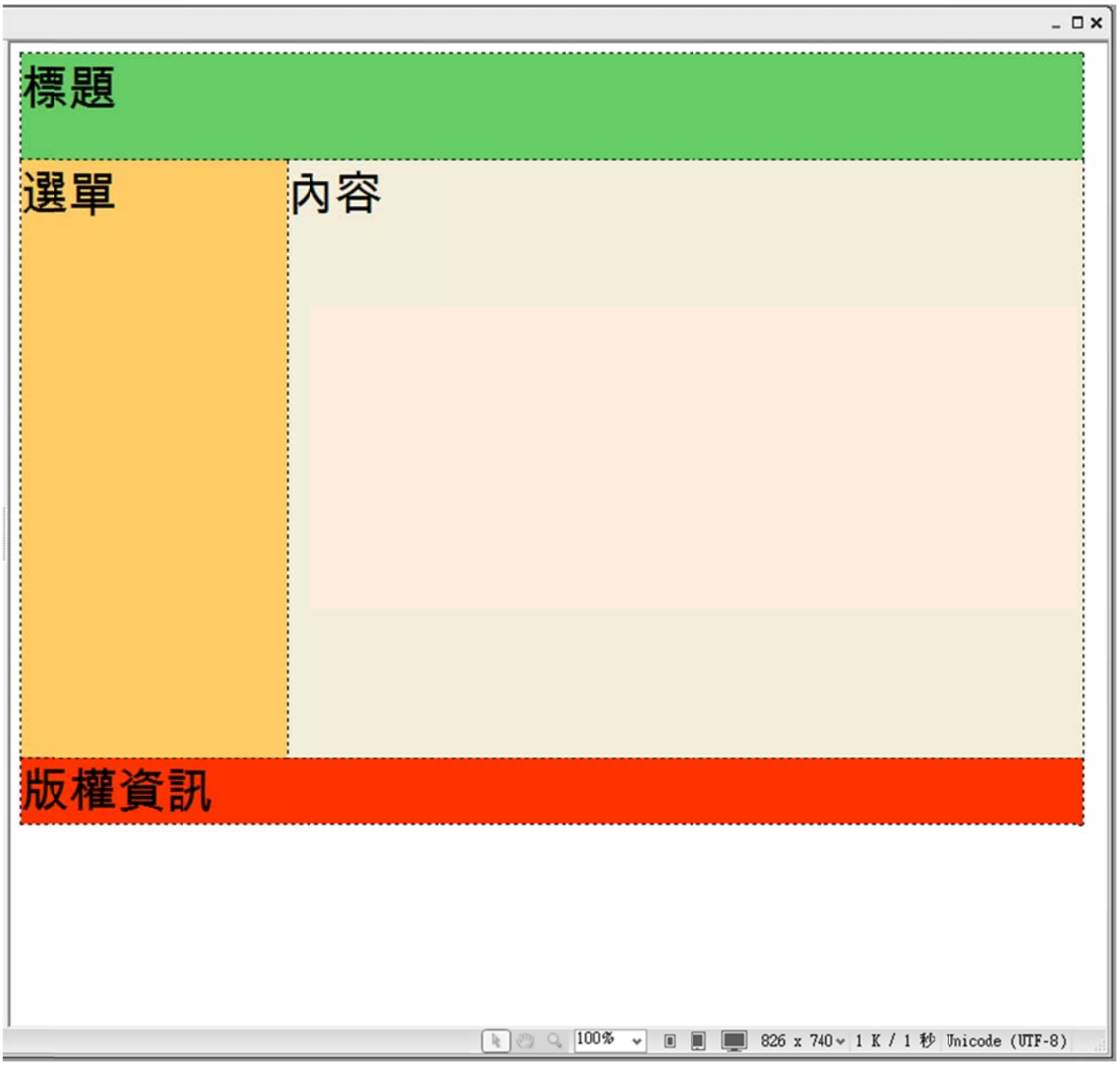
4.11 DIV 與 CSS 佈局設計

多年前採用框架來佈局是很熱門的一種方式，但是到了現代，一個好的網頁佈局應該要採用 DIV 與 CSS，如下所示：

C:\Users\Jerry\Desktop\example.htm

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            * { /* 所有元素皆套用些樣式 */
                font-family: Arial Unicode MS;
                font-size: 36px;
            }
            #header {
                width: 800px;
                height: 80px;
                background-color: #6C6;
            }
            #sidebar {
                float: left; /* 向左浮動對齊 */
                width: 200px;
                height: 450px;
                background-color: #FC6;
            }
            #content {
                float: left;
                width: 600px;
                height: 450px;
                background-color: #F3EEDA;
            }
            #footer {
                clear: both; /* 清除兩邊環繞 */
                width: 800px;
                height: 50px;
                background-color: #F30;
            }
        </style>
    </head>
    <body>
        <div id="header">標題</div>
        <div id="sidebar">選單</div>
        <div id="content">內容</div>
        <div id="footer">版權資訊</div>
    </body>
</html>
```

以上程式碼的執行結果如下：



5 CSS 語法

5.1 背景屬性

屬性	屬性值	描述
background-color	#RRGGBB、#RGB、color	設定背景色彩
background-image	url(example.png)	設定背景圖片
background-repeat	repeat-x、repeat-y、no-repeat	設定是否重覆背景圖片
background-attachment	scroll、fixed	設定是否固定背景圖片的位置
background-position	26px 32px	設定背景圖片的位置
background	以上全部	一次設定所有屬性

5.2 邊框屬性

屬性	屬性值	描述
$\text{border} - \begin{pmatrix} top \\ left \\ right \\ bottom \end{pmatrix} - \begin{pmatrix} color \\ style \\ width \end{pmatrix}$	style： solid、 dashed、 dotted、 none	設定邊框的 色彩、 樣式、 寬度
$outline - \begin{pmatrix} color \\ style \\ width \end{pmatrix}$	同上	設定輪廓(邊框外圍)的 色彩、 樣式、 寬度

5.3 文字屬性

屬性	屬性值	描述
color	#RRGGBB、#RGB、color	設定文字色彩
letter-spacing	17px、17em、17%	設定文字間距
line-height	17px、17em、17%	設定行高
text-align	left、right、center	設定文字對齊方式

5.4 字體屬性

屬性	屬性值	描述
font-family	"Arial Unicode MS","微軟正黑體","新細明體"	設定字型優先順序
font-size	17px、17em、17%	設定字體大小
font-style	italic	設定斜體
font-weight	bold	設定粗體
font	以上全部	一次設定所有屬性

5.5 邊距屬性

屬性	屬性值	描述
$\begin{matrix} margin \\ padding \end{matrix} - \begin{pmatrix} top \\ left \\ right \\ bottom \end{pmatrix}$	17px、17em、17%	設定邊距的寬度

讓元素水平置中的方法：margin: 0 auto;　/* 上下邊距為 0，左右邊距自動調整 */

5.6 尺寸屬性

屬性	屬性值	描述
height	17px、17em、17%	設定元素高度
width	17px、17em、17%	設定元素寬度

5.7 定位屬性

屬性	屬性值	描述
clear	left、right、both	清除左、右、兩側環繞
display	inline、block	將元素視為區塊或行內
float	left、right	向左、右浮動對齊
vertical-align	17px、17em、17%	設定元素的垂直對齊方式

6 HTML5 與 CSS3 的變革

6.1 用 CSS 取代 HTML 的樣式設定標籤

W3C 鼓勵網頁程式設計師用 CSS 設定樣式，取代傳統上直接用 HTML 屬性設定樣式的方法。在新版的 HTML5 與 CSS3 中，W3C 捨棄了許多 HTML 屬性，但增加了多種 CSS 屬性，希望可以讓網頁的樣式設定全部改用 CSS 編寫。

6.2 更多元的排版與更豐富的特效

新版的 CSS3 提供了更多元的樣式選擇，例如：多欄版面配置、圓角化與增加陰影、簡單動畫……等等，讓網頁開發更為方便。但這些元素都無法在舊版的瀏覽器中使用。

6.3 Canvas、Audio、Video

Canvas 元素可以建立一個繪圖區域，搭配其他腳本語言(如 JavaScript)即可即時地繪製出各式圖形。Audio 元素可以讓使用者不需安裝任何瀏覽器外掛程式(如 Windows Media Player)即可在網頁中播放音樂。Video 元素則可以讓使用者不需安裝任何瀏覽器外掛程式即可在網頁中播放影片。但這些元素都無法在舊版的瀏覽器中使用。

6.4 語義化標籤取代 Div 標籤

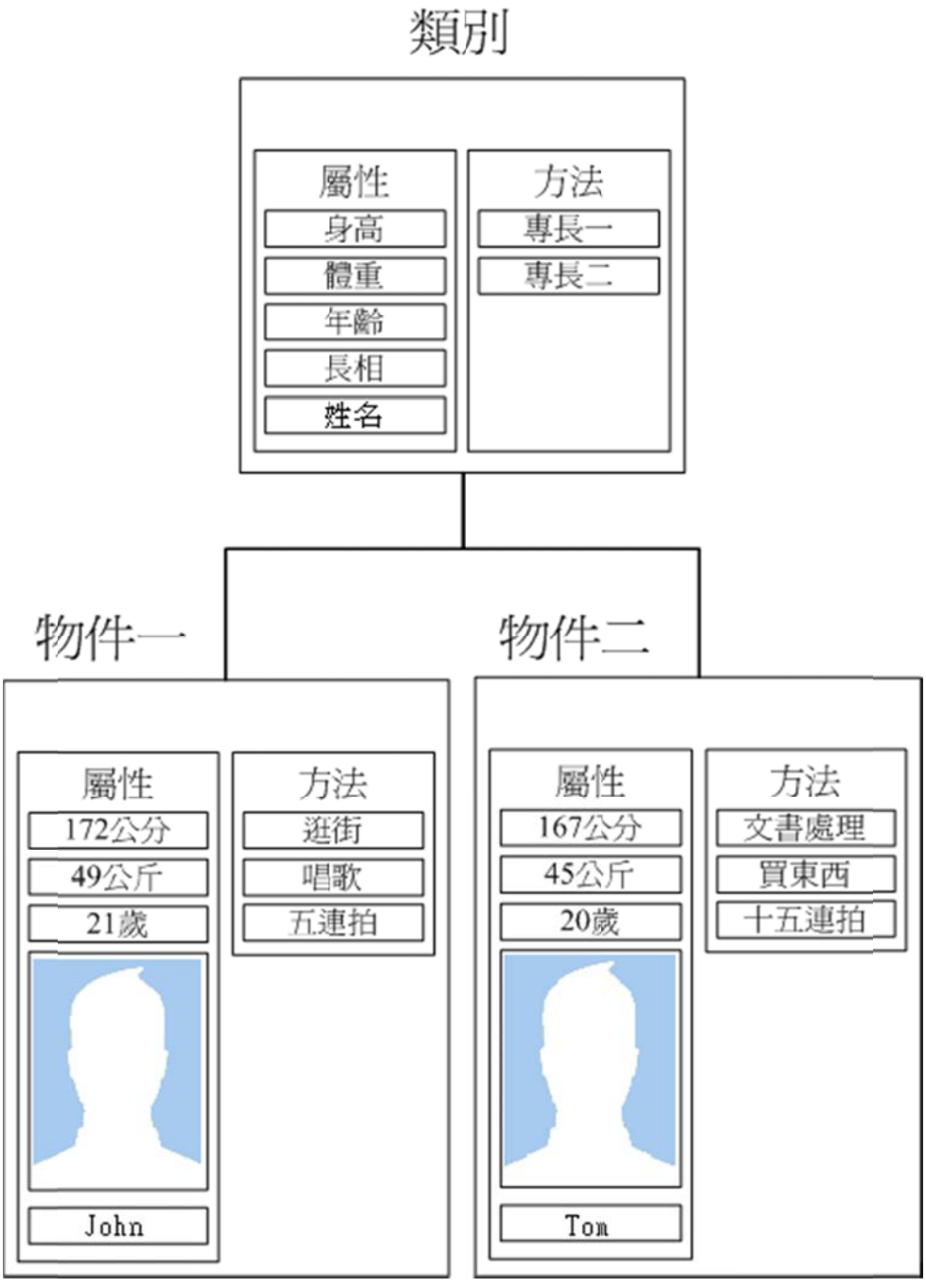
在 HTML5 中，新增了 Article、Aside、Footer、Header、Section、Nav 等標籤取代傳統上直接用 Div 標籤分割版面的做法。其功能與 Div 標籤相似，但語義化的特性使其易更於編寫。不過現存的 Div 標籤並未因此被捨棄，網頁開發者仍可使用 Div 標籤達成版面的分割。

7 JavaScript 腳本語言

7.1 ECMAScript 與 JavaScript

1996 年，網景(Netscape)為自家瀏覽器開發了一種互動性的腳本語言，稱為 JavaScript，緊接著，微軟(Microsoft)也為自家瀏覽器開發了另一種與 JavaScript 極為相似的 Jscript 腳本語言來應戰。由於兩種語言語法略有差異，不利跨瀏覽器之網頁開發，1997 年 ECMA 為其制定一套標準，稱為 ECMAScript，以統一兩種語言的規格。目前，網景與微軟這兩間公司所開發的 JavaScript 與 Jscript 都遵守該規範，但同時也各自擴充自家語言的功能。

7.2 JavaScript 的物件導向



大部份的物件導向程式語言，如 C++、Java、C#，都是屬於類別基礎(Class-based)的物件導向程式語言，其強調類別(Class)的畫分，並針對類別建立實例(Instance)，如上圖所示。但 JavaScript 是屬於原型基礎(Prototype-based) 的物件導向程式語言，其不強調類別與實例的差異，類別事實上就是物件，在 JavaScript 裡使用物件並不需事先設定類別即可直接建立。

7.3 變數

JavaScript 與 C 語言一樣，具有輸出、判斷、計算、建立函式等功能，其語法也與 C 語言有許多共通之處，例如：用一個分號「;」表示每一行程式的結束，用「//」、「/* ... */」表示註解。唯 JavaScript 在變數的使用上與 C 語言有很大的不同，JavaScript 在宣告一個變數時不需指定該變數的資料型態，只需使用 var 指令建立即可，程式執行時會自動依據變數內的值來設定其資料型態。事實上，就算一個變數沒有事先宣告，也可以直接使用，例如：

```
//宣告一個名為"str"的變數
var str;

//宣告的同時亦指定其值為"Hello World !!"，該變數將自動被設為字串資料型態
var str="Hello World !!";

//就算未經宣告仍可以直接使用
str="Hello World !!";
```

JavaScript 變數命名的規定為：一、不能使用保留字。二、變數名稱的開頭需為英文字元或「_」符號。三、變數名稱的其他部份可為英文字元、數字或「_」符號。四、區分大小寫。另外，JavaScript 的資料型態分為：整數、浮點數、字串、布林值、物件、陣列、Null、Undefined，雖然 JavaScript 不強調變數的資料型態，但是當我們將多個變數混合運算時，變數的資料型態還是會影響到運算的結果，因此仍需留意。JavaScript 的運算字大致與 C 語言相同，在此不再多做敘述。

7.4 基本輸出：write、alert、confirm

JavaScript 的輸出方式最簡單的便是使用 document 物件的 write 方法，它會在將訊息直接顯示在網頁上。另外還有 alert 函數，會跳出對話視窗，並將訊息顯示在對話視窗中。最後一個 confirm 函數則是會像 alert 函數一樣出現對話視窗並顯示訊息於其中，但多了「是」、「否」兩個選項，提供使用者做選擇。詳見以下範例：

```
//document.write() example
var name="Patty";
document.write("Hi "+name+" !!");

//alert() example
var name="Patty";
```

```
alert("Hi "+name+" !!");

//confirm() example
if(confirm("Are you sure you want to quit?")){
	alert("I don't let you go !!");
}
else{
	alert("Good choice.");
}
```

7.5 基本屬性：value、innerHTML、style

我們可以透過 document 物件的 getElementById()方法取得 DOM 節點樹上特定 ID 屬性值的元素，另外，document 物件的 getElementsByTagName()方法則可以取得特定元素名稱的元素。取得元素後，我們可以再使用 value 或 innerHTML 屬性來存取它的內容，例如：

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <textarea id="eg">example</textarea>
    <script type="text/javascript">
        //用getElementById()取得元素
        var objA=document.getElementById("eg");
        alert(objA.value);
        objA.value="hello";
        //用getElementsByTagName()取得元素
        var objB=document.getElementsByTagName("textarea");
        alert(objB[0].value);
        objB[0].value="example";
    </script>
</body>
</html>
```

getElementsByTagName 取得的是一個符合要求的元素的清單，我們可以將之視為一個陣列，並輸入索引值以取得其各別的元素。value 屬性是用來取得元素中不含 HTML 代碼的內容，而 innerHTML 屬性則是用來取得元素中含 HTML 代碼的內容，但有時它們取得的值皆相同。另外還有一個 style 屬性，它可以用來存取 CSS 樣式，例如：

```
<!DOCTYPE html>
<html>
```

```
<head>
</head>
<body>
    <textarea id="eg">example</textarea>
    <script type="text/javascript">
        //用getElementById()取得元素
        var objA=document.getElementById("eg");
        objA.style.color="pink";
        //用getElementsByTagName()取得元素
        var objB=document.getElementsByTagName("textarea");
        objB[0].style.width="200px";
    </script>
</body>
</html>
```

需特別注意 CSS 樣式屬性轉換成 style 物件屬性時，若原本的屬性名稱內有「-」符號需省略，並將後面的英文字首改成大寫，例如 background-color 這一 CSS 樣式屬性轉換成 style 物件屬性後會變成 backgroundColor。

7.6 流程控制

JavaScript 的流程控制大致與 C 語言相同，在此不再多做敘述。

條件控制	if/else
	switch
	?: 範例：time = (hour >= 12) ? "PM" : "AM";
迴圈控制	for
	while
	do/while

7.7 函式

JavaScript 的函式撰寫大致與 C 語言相同，在此不再多做敘述。範例：

```
<script type="text/javascript">
    function setStyle(n){
        if (document.getElementById(n).value == "請輸入帳號" || document.getElementById(n).
        value == "請輸入密碼")
            document.getElementById(n).value = "";
        else if (document.getElementById(n).value == ""){
            if (n == "username")
```

```
                document.getElementById(n).value = "請輸入帳號";
            else
                document.getElementById(n).value = "請輸入密碼";
        }
    }
</script>
<form>
    <input type="text" id="username" onfocus="setStyle(this.id)" onblur="setStyle(this.id)" value="請
    輸入帳號" />
    <input type="text" id="password" onfocus="setStyle(this.id)" onblur="setStyle(this.id)" value="請
    輸入密碼" />
    <input type="submit" value="登入" />
</form>
```

7.8 Array 物件

使用 JavaScript 建立陣列有以下三種方法：

```
//使用 Array()建構函數，事後指定各項陣列值
var classmates=new Array();
classmates[0]= "Tom";
classmates[1]= "Merry";
classmates[2]= "Andy";

//使用 Array()建構函數，同時指定各項陣列值
var classmates=new Array("Tom","Merry","Andy");

//不使用 Array()建構函數
var classmates=["Tom","Merry","Andy"];
```

new 運算子對 JavaScript 是非常中要的，若要使用建構函數來建立物件，則前面都必需加上 new 這個運算子。所謂的「建構函數」，指的就是用來建立物件的函數，例如，建立陣列所使用的建構函數就是 Array()。JavaScript 內建的 Array 物件提供了豐富的屬性和方法，常見的有：

length	取得陣列的元素個數
concat(array)	合併陣列，將參數中的陣列接到目前陣列之後
join()	依序輸出陣列中的每一個元素
pop()	刪去陣列的最後一個元素
push("item")	在陣列的結尾增加一個元素，參數為該元素之值
reverse()	將陣列中的元素順序顛倒排列
sort()	將陣列中的元素按字元順序排列

7.9 Date 物件

Date 物件可以取得系統時間，使用方法如下：

```
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript">
       function time(){
           //建立Array物件
           var weekday = new Array("星期日","星期一","星期二","星期三",
           "星期四","星期五","星期六");
           //建立Date物件
           var today = new Date();
           //透過getFullYear方法取得年份
           var output = today.getFullYear() + "/";
           //透過getMonth方法取得月份，0~11分別代表一至十二月
           output += ( today.getMonth() + 1 ) + "/";
           //透過getDate方法取得日期
           output += today.getDate() + "<br/>";
           //輸出系統日期
           document.write( "系統日期：" + output );
           //透過getHours方法取得小時
           output = today.getHours() + ":";
           //透過getMinutes方法取得分鐘
           output += today.getMinutes() + ":";
           //透過getSeconds方法取得秒數
           output += today.getSeconds() + "<br/>";
           //輸出系統時間
           document.write( "系統時間：" + output );
           //透過getDay方法取得星期值，0~6分別代表星期日到六
           output = today.getDay();
           //輸出系統星期值
           document.write( weekday[output] );
       }
    </script>
</head>
<body onload="time()">
</body>
</html>
```

7.10 自訂物件

除了使用上述 JavaScript 內建的物件，我們也可以自己建立物件。要建立自訂物件，有以下三種方法：

直接建立物件	var objCard = {
	name : "Tom",
	age : "17",
	phone : "58585888",
	email : "tom@gmail.com"
	};

用 Object()建構函數建立物件	var objCard = new Object(); objCard.name = "Tom"; objCard.age = "17"; objCard.phone = "58585888"; objCard.email = "tom@gmail.com";

自行撰寫建構函數建立物件，方便重複使用該結構	function nameCard(name, age, phone, email){
	this.name = name;
	this.age = age;
	this.phone = phone;
	this.email = email;
	}
	var objCardA = new nameCard("Tom","17","58585888","tom@gmail.com");
	var objCardB = new nameCard();
	objCardB.name = "Tom";
	objCardB.age = "17";
	objCardB.phone = "58585888";
	objCardB.email = "tom@gmail.com";

剛剛我們都只有建立物件的屬性，當然，我們也可以建立物件的方法，只要設定該方法對應到的函數即可：

```
function nameCard(name, age, phone, email){
    this.name = name;
    this.age = age;
    this.phone = phone;
    this.email = email;
    this.print = printCard;
}
function printCard(){
    document.write( "姓名：" + this.name + "<br/>" );
    document.write( "年齡：" + this.age + "<br/>" );
```

```
        document.write( "電話：" + this.phone + "<br/>" );
        document.write( "郵件：" + this.email + "<br/>" );
}
var objCard = new nameCard();
objCard.name = "Tom";
objCard.age = "17";
objCard.phone = "58585888";
objCard.email = "tom@gmail.com";
objCard.print();
```

學到這裡，相信大家對 JavaScript 應該已經有一些認識了，接下來，如果你對設計客戶端動態網頁有興趣，建議你去學 jQuery，它是一個功能強大的 JavaScript 函式庫，可以讓你輕易地建立出豐富多樣的網頁。

8 基礎 PHP

8.1 安裝 XAMPP

XAMPP 是一套結合 Apache 網頁伺服器、PHP 伺服器、MySQL 伺服器、phpmyadmin 資料庫管理工具及其他常用元件的軟體，在撰寫 PHP 程式碼之前，請先至 www.apachefriends.org/zh_tw/xampp-windows.html 下載最新版本的 XAMPP，並完成安裝。

安裝完成後，我們可以在「開始／所有程式／XAMPP」中找到 XAMPP 控制面版，請到控制面版內確認你要使用的服務已經開啟。接著，前往 http://localhost 或 http://127.0.0.1，我們可以看到一個 XAMPP 預先幫我們建立好的網頁。

往後我們只要將完成的網頁放到安裝目錄下的「\xampp\htdocs」資料夾，就可以在 http://localhost 內開啟它。例如我們在 htdocs 資料夾中放一個 example.htm，則對應到的網址就是 http://localhost/example.htm。若我們在 htdocs 資料夾中再建立一個名為 test 的資料夾，並放入 example.htm，則對應到的網址就是 http://localhost/test/example.htm。

8.2 變數

PHP 與 JavaScript、C 語言一樣，具有輸出、判斷、計算、建立函式等功能，其語法也與 JavaScript、C 語言有許多共通之處，例如：用一個分號「;」表示每一行程式的結束，用「//」、「/* ... */」表示註解。在變數的使用上，PHP 與 JavaScript 相似，在宣告一個變數時不需指定該變數的資料型態，程式執行時會自動依據變數內的值來設定其資料型態，並且一個變數就算未經宣告仍可以直接使用。不過 PHP 的變數使用也有一些與 JavaScript 不同之處，那就是 PHP 的每個變數名稱前需加上「$」符號作為識別。請看以下範例：

```
<?php
//指定變數"$str"之值為"Hello World !!"，該變數將自動被設為字串資料型態
```

```
$str="Hello World !!";

//指定變數"$num"之值為"17"，該變數將自動被設為整數資料型態
$num=17;
?>
```

PHP 變數命名的規定為與 JavaScript 相同，在此不再多做敘述。而 PHP 的資料型態分為：整數、浮點數、字串、布林值、物件、陣列、資源、Null，PHP 與 JavaScript 一樣，雖然都不強調變數的資料型態，但是當我們將多個變數混合運算時，變數的資料型態還是會影響到運算的結果，因此仍需留意。PHP 的運算字大致與 JavaScript、C 語言相同，在此不再多做敘述。

8.3 輸出

在 PHP 裡，我們常用下列四種方法輸出訊息：

echo	直接輸出	<?php $name="Patty"; echo "Hi ".$name." !!"; ?>
die()	輸出指定訊息並終止執行當前的程式碼，常用於輸出錯誤訊息	<?php isset($str) or die("變數$str 不存在"); ?>
printf()	指定資料型態輸出	<?php $name="Patty"; printf("Hi %s !!", $name); ?>
sprint()	組合字串但不輸出	<?php $name="Patty"; $str=sprintf("Hi %s !!", $name); echo $str; ?>

其中，printf()、sprintf()的轉換規格與 C 語言相同，在此不再多做敘述。

%s	字串
%c	字元
%d	十進制
%b	二進制
%f	浮點數

8.4 流程控制

PHP 的流程控制大致與 JavaScript、C 語言相同，在此不再多做敘述。

條件控制	if/else
	switch
	?: 範例：$time = ($hour >= 12) ? "PM" : "AM";
迴圈控制	for
	while
	do/while

8.5 陣列

在 PHP 中使用陣列有以下幾種方法：

以數字為索引值	<?php $name = array("Chen", "Lin", "Jang", "Lee", "Wang"); ?>
	<?php $name[0] = "Chen"; $name[1] = "Lin"; $name[2] = "Jang"; $name[3] = "Lee"; $name[4] = "Wang"; ?>
以字串為索引值	<?php $name = array("a" => "Chen", "b" => "Lin", "c" => "Jang", "d" => "Lee", "e" => "Wang"); ?>
	<?php $name["a"] = "Chen"; $name["b"] = "Lin"; $name["c"] = "Jang"; $name["d"] = "Lee"; $name["e"] = "Wang"; ?>

8.6 函式

PHP 的函式撰寫大致與 JavaScript、C 語言相同，在此不再多做敘述。請看以下範例，在這個範例中，我們會發現 PHP 的流程控制可以直接搭配 HTML 程式碼進行操作，不像一般客戶端語言，只能透過輸出 HTML 程式碼的方式，間接完成操作。

```
<?php
    function login($username, $password){
        if($username == "ckefgisc" && $password == "173226"){
            $login_success = true;
            return $login_success;
        }
        else{
            $login_success = false;
            return $login_success;
        }
    }
?>
<?php if($login_success){ ?>
    <p>登入成功</p>
<?php } else { ?>
    <p>登入失敗</p>
<?php } ?>
```

8.7 COOKIE 與 SESSION

COOKIE 與 SESSION 都是扮演著記錄使用者資訊的角色。我們只要登入一個網站中的其中一個網頁，便可以在該網站中自由瀏覽，甚至關閉瀏覽器後，網站也能記住我們的帳密，讓我們下次再使用的時候，不用重新登入，這靠的全是 COOKIE 與 SESSION。

不過 COOKIE 與 SESSION 仍然有一些差別，COOKIE 是將使用者資訊儲存在我們電腦中，SESSION 則是將使用者資訊儲存在伺服器中，並且當使用者關閉瀏覽器後，SESSION 也會跟著消失，故我們常說 SESSION 的安全性較 COOKIE 高。事實上，當一個 SESSION 建立的同時，在我們的電腦中也會產生一個 COOKIE，不過它是用來記錄該 SESSION 的 ID，如此一來才能判斷哪個 SESSION 是隸屬於哪個使用者。

設置 COOKIE	setcookie("名稱","內容","期限");	setcookie("time",date("H:i:s"),time()+3600);
	$_COOKIE['名稱']="內容";	$_COOKIE['time']=date("H:i:s");
顯示 COOKIE	$_COOKIE['名稱'];	echo $_COOKIE['time'];
刪除 COOKIE	setcookie("名稱","內容","期限");	setcookie("time","",time()-60);
設置 SESSION	session_start(); $_SESSION['名稱']="內容";	session_start(); $_SESSION['time']=date("H:i:s");

顯示 SESSION	session_start(); echo $_SESSION['名稱'];	session_start(); echo $_SESSION['time'];
刪除 SESSION	session_start(); session_destroy();	session_start(); session_destroy();
	session_start(); unset($_SESSION['名稱']);	session_start(); unset($_SESSION['time']);

$_COOKIE、$_SESSION 都是 PHP 的預定義變數。date()可以取得並格式化目前的時間，time()同樣也可以取得目前的時間，不過我們得到的是距離 1970 年 1 月 1 日 0 時 0 分 0 秒的秒數。想要刪除 COOKIE，只要將 setcookie()的期限設為過去的時間，這樣這個 COOKIE 自然就失效了。在使用 SESSION 時需注意一定要加上 session_start()，且 session_start()之前不能有輸出。

8.8 GET 與 POST

GET 與 POST 是兩種表單元件傳遞與接收的方法，詳見以下範例。

GET 使用範例：

1.php

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
    <form name="example" action="2.php" method="get">
    姓名：<input type="text" name="name"/><br/>
    性別：<input type="radio" name="sex" value="male"/>男
    <input type="radio" name="sex" value="female"/>女<br/>
    <input type="submit" value="送出"/>
    </form>
</body>
```

2.php

```
<?php
    echo $_GET['name'].",".$_GET['sex'];
?>
```

POST 使用範例：

1.php

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
```

	<form name="example" action="2.php" method="post"> 姓名：<input type="text" name="name"/> 性別：<input type="radio" name="sex" value="male"/>男 <input type="radio" name="sex" value="female"/>女 <input type="submit" value="送出"/> </form> </body>
2.php	<?php echo $_POST['name'].",".$_POST['sex']; ?>

使用 GET 時我們要將 form 元素的 method 屬性值設為 GET，相對的，使用 POST 時屬性值就要設為 POST。兩個方法的差別在於，GET 會將傳遞的值放置在網址中，POST 則會將傳遞的值放置在 HTTP 檔頭中，故 POST 較 GET 安全且可傳遞的資料夾較大。

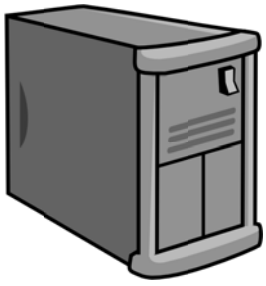
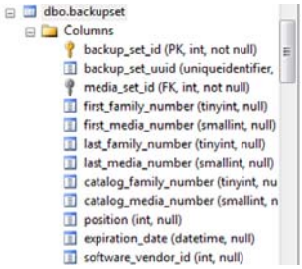
事實上，使用 GET 時在網址中出現的那一串字串就是所謂的 URL 參數，其格式為「網址?變數一的名稱=變數一的值&變數二的名稱=變數二的值&……」。因此你應該會看到「http://localhost/2.php?name=YourName&sex=YourSex」若你在表單輸入中文，由於在傳遞前會先經過編碼，所以你看到的參數會很像亂碼。

思考：試說明如何間接的將一個 PHP 變數與一個 JavaScript 變數之值相互傳遞。

9 MySQL 資料庫

9.1 關聯式資料庫

關聯式資料庫是目前極為普遍的一種資料庫形式，其資料庫的結構為：

伺服器(Server)	資料庫(Database)	資料表(Table)	欄位(Column)
服務提供者	多張表格形成之集合	一張表格	表格中的每項標題
			

MySQL 即為一種關聯式資料庫，在開始學習 MySQL 指令之前，請先安裝 SQLyog，這是一款圖形化的 MySQL 資料庫管理軟體，我們將用它來幫助我們熟悉 MySQL 指令。當然，如果你的程度夠好，也可以使用 MySQL 內建的純文字管理工具。

9.2 主鍵與連外鍵(Primary Key & Foreign Key)

這裡有兩份資料表，一份是「學生基本資料」資料表，另一份是「郵遞區號」資料表。我們將郵遞區號另外建置成一個表格，是為了降低資料的重複性，節省資料庫空間。

學生基本資料

學號	姓名	年齡	居住鄉鎮
123456	梅友仁	27	鶯歌鎮
123457	是好人	15	鶯歌鎮
123458	路人甲	29	中正區

郵遞區號

居住鄉鎮	郵遞區號
鶯歌鎮	239
中正區	100

資料表中每一項藍底白字的標題就是一個「欄位」，在關聯式資料庫中，有兩種特殊欄位，就是「主鍵」與「連外鍵」。主鍵是被用來辨識資料表中唯一的記錄，在「學生基本資料」資料表中，應該將「學號」欄位作為該資料表的主鍵，如此一來，我們可以藉由學號查詢到唯一的一筆學生資料。其他欄位因為都可能有重複的值，故不可作為主鍵。連外鍵是被用來為兩份資料表建立連結，凡是對應到其他資料表中主鍵的欄位便被稱為連外鍵。在上面這個例子中，「學生基本資料」資料表與「郵遞區號」資料表以「居住鄉鎮」作為連結，「居住鄉鎮」欄位就是一個連外鍵。若一份資料表要連結到多份資料表，則可能建立多個連外鍵。

9.3 SELECT 指令

一個資料庫中可能會有多份資料表，這時我們就要用 SELECT 指令選擇我們想要查詢的資料表。以下為 SELECT 指令的語法，請注意，在 MySQL 中，欄位和資料表的名稱不是用「'」符號包圍，而是用 Tab 鍵上方的「`」符號包圍，「'」符號是用來包圍字串的。另外，若欄位和資料表的名稱並沒有使用到 MySQL 的保留字，則包圍的「`」符號可以省略。若我們想查詢一份資料表中的所有欄位，只要將語法中的「`欄位名稱`」改成「*」即可。

```
SELECT  `欄位名稱`  FROM  `資料表名稱`
```

9.4 LIMIT 子句

若該資料表中的記錄筆數很多，我們可以加上 LIMIT 子句來限制傳回的記錄筆數，語法如下。例如輸入「LIMIT 0,30」，即表示要從第 0 筆記錄開始，傳回 30 筆記錄。

```
SELECT  `欄位名稱`  FROM  `資料表名稱`
LIMIT  起始記錄  ,  傳回的記錄筆數
```

9.5 WHERE 子句

WHERE 子句可以讓我們篩選查詢的資料，有點類似搜尋引擎的概念。例如我們只想取得「Addr」欄位的值等於「台北市」的記錄，則可在輸入指令時加上：

```
WHERE  `Addr`  =  '台北市'
```

請注意包圍名稱的「`」與包圍字串的「'」，不要打錯。另外，MySQL 支援的運算子有以下幾種：

等於	=　　//注意，不是「==」!
不等於	!=或<>
大於	>
大於等於	>=
小於	<
小於等於	<=
字串的比對	LIKE

LIKE 運算子要搭配萬用字元「_」、「%」來使用。「_」代表一個任意的字元，「%」則是代表一串任意的字串，例如我們只想取得「Addr」欄位的值等於「XX 市」的記錄，則可在輸入指令時加上：

```
WHERE  `Addr`  LIKE  '%市'
```

我們也可以使用「AND」與「OR」進行多個條件的查詢，例如我們只想取得「Addr」欄位的值等於「XX 市」，並且「Age」欄位的值小於等於「18」的記錄，則可在輸入指令時加上：

```
WHERE  `Addr`  LIKE  '%市'  AND  `Age`  <=  18
```

9.6 ORDER BY 子句

ORDER BY 子句可以替某個特定的欄位進行排序，語法如下。「ASC」是「遞增」的意思，相對的，「DESC」就是「遞減」的意思。

```
SELECT  `欄位名稱`  FROM  `資料表名稱`
ORDER  BY  `欄位名稱`  ASC/DESC
```

9.7 INSERT 指令

INSERT 指令可以插入一筆記錄到資料庫，其語法如下：

```
INSERT INTO `資料表名稱` ( `欄位一` , `欄位二` , `欄位三` )
VALUES ( '值一' , '值二' , '值三' )
```

9.8 UPDATE 指令

UPDATE 指令可以更新舊有的記錄，其語法如下：

```
UPDATE `資料表名稱` SET `欄位一` = '值一' , `欄位二` = '值二'
WHERE `主鍵` = '主鍵值'
```

其中的 WHERE 子句也可以使用其他的 WHERE 條件判斷語句，詳見 9.5 之說明。若使用 UPDATE 指令時未加上 WHERE 子句，則會更新所有記錄。為避免這種錯誤發生，我們可以再指令後面加上「LIMIT 1」來限制資料庫只能對一筆記錄進行操作。

9.9 DELETE 指令

DELETE 指令可以刪除舊有的記錄，其語法如下：

```
DELETE FROM `資料表名稱`
WHERE `主鍵` = '主鍵值'
```

和 UPDATE 指令一樣，WHERE 子句可以使用任何 WHERE 條件判斷語句，若忘了加上 WHERE 子句，所有記錄都會被刪除，為避免錯誤可以加上「LIMIT 1」子句加以限制。

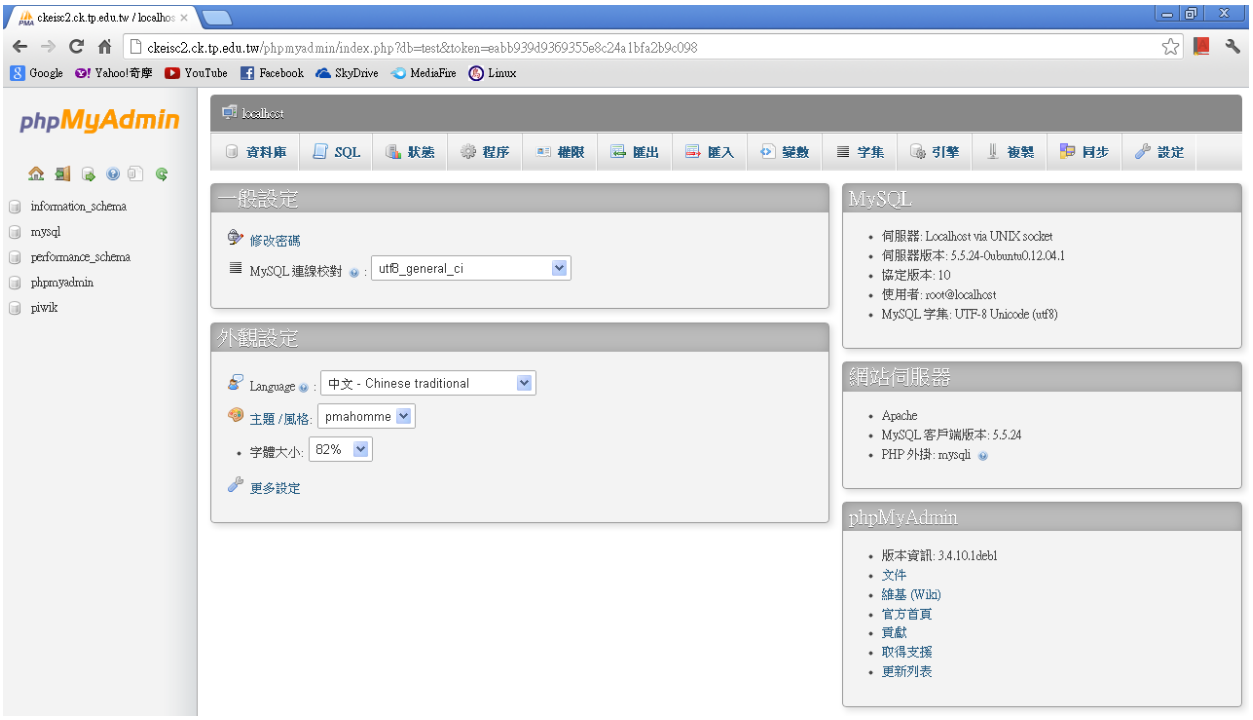
9.10 MySQL 的資料型態

MySQL 就和一般的程式語言一樣，每個欄位都有其資料型態，常見的 MySQL 資料型態如下：

數值	INT	-2E+31～2E+31-1
	FLOAT	-3.4E+38～3.4E+38
	DOUBLE	-2.2E+308～2.2E+308
文字	VARCHAR	1～255 字元
時間	TIMESTAMP	1970 年～2037 年

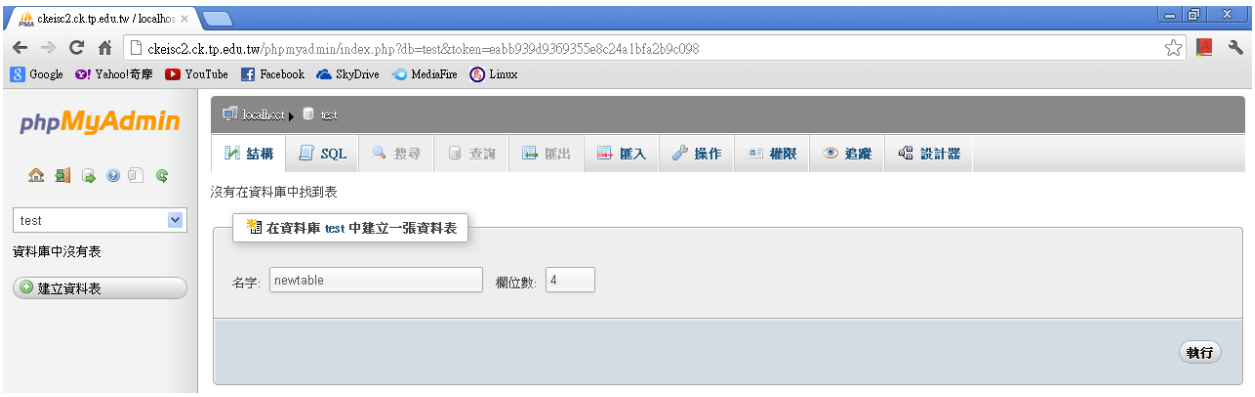
9.11 phpMyAdmin 的操作

①進入 phpMyAdmin 管理系統：



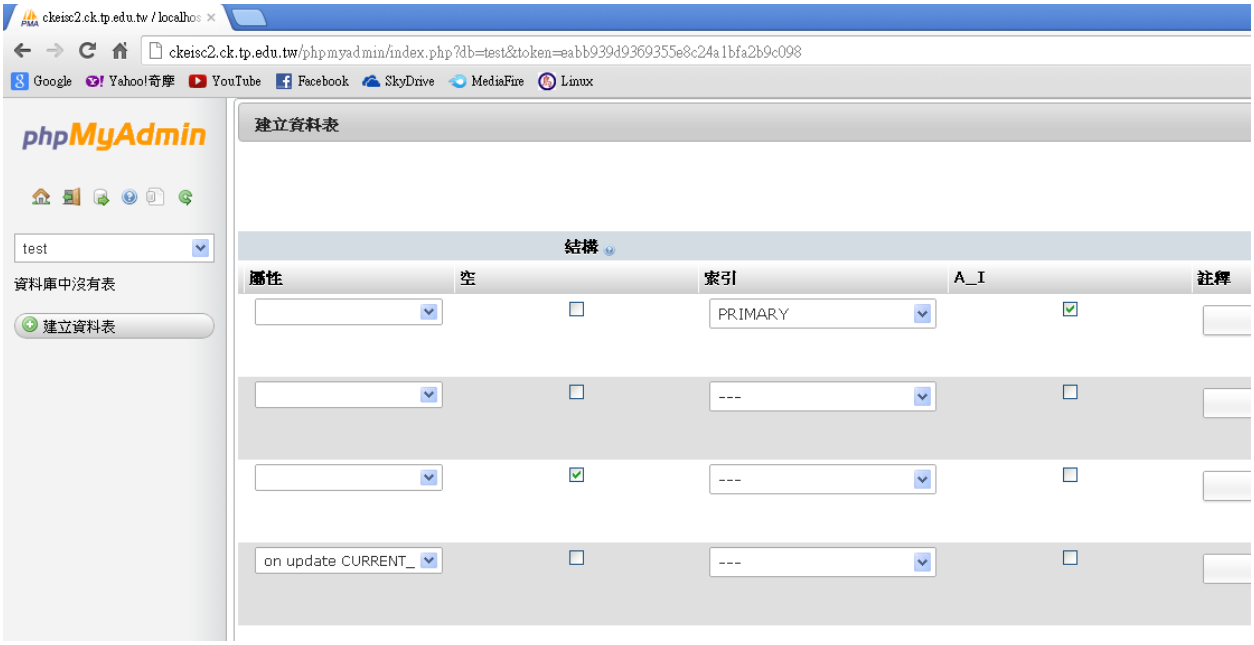
②建立資料庫：

③建立資料表：



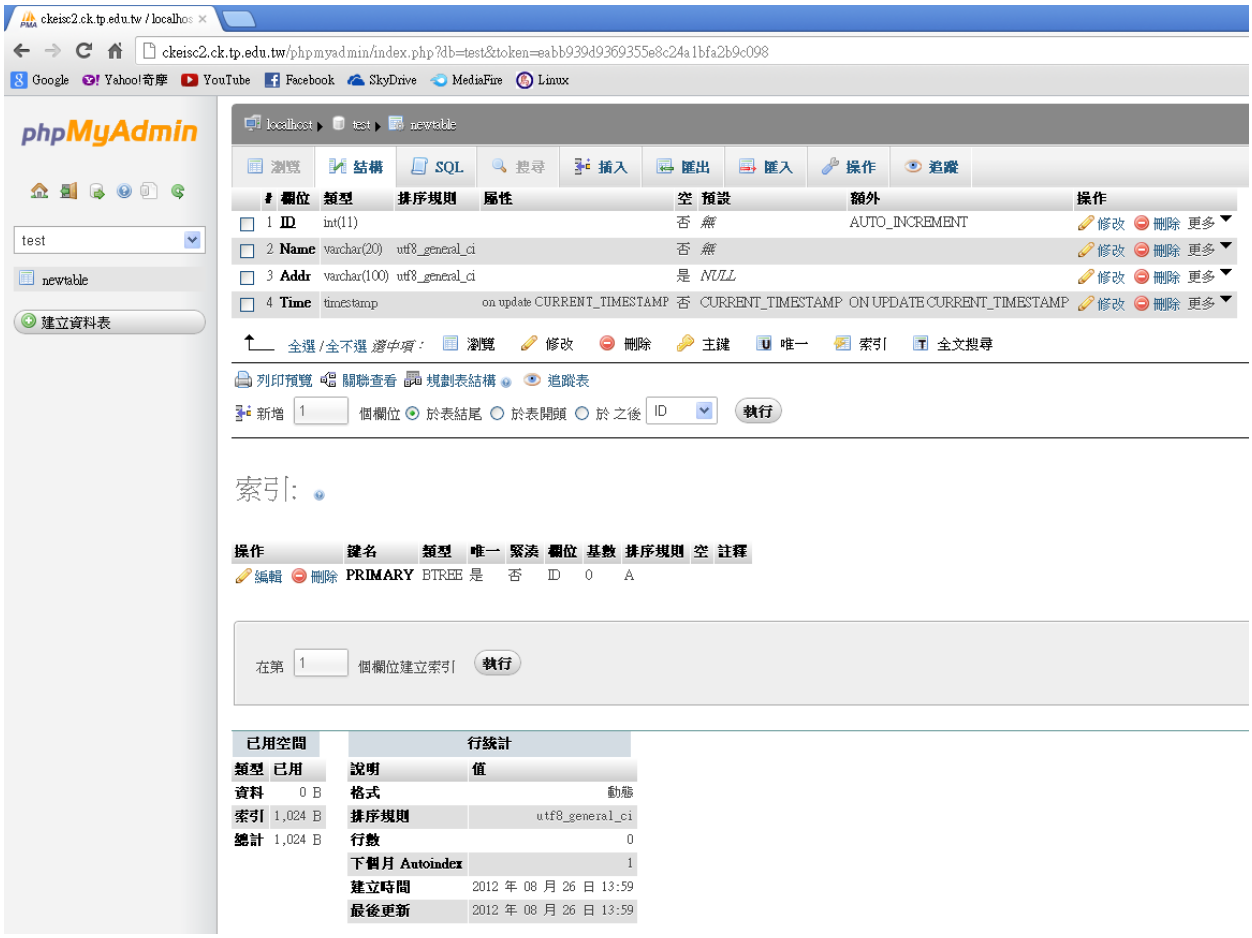
④建立欄位：

phpMyAdmin
建立資料表
資料資料表名稱:
newtable
test
資料庫中沒有表
建立資料表
欄位
類型
長度/值1
預設2
排序規則
ID
INT
無
utf8_general_ci
Name
VARCHAR
20
無
utf8_general_ci
Addr
VARCHAR
100
無
utf8_general_ci
Time
TIMESTAMP
CURRENT_TIMESTAMP
utf8_general_ci
表註釋:
儲存引擎:
MyISAM
排序規則:
utf8_general_ci
分區定義:
儲存
或 新增
1
個欄位
執行
1 如欄位類型是"enum"或"set"，請使用以下格式輸入：'a','b','c'...
如果需要輸入反斜槓("\")或單引號("")，請在前面加上反斜槓(如 '\\xyz' 或 'a\'b')
2 對於預設值，請只輸入單個值，不要加反斜槓或引號，請用此格式：a

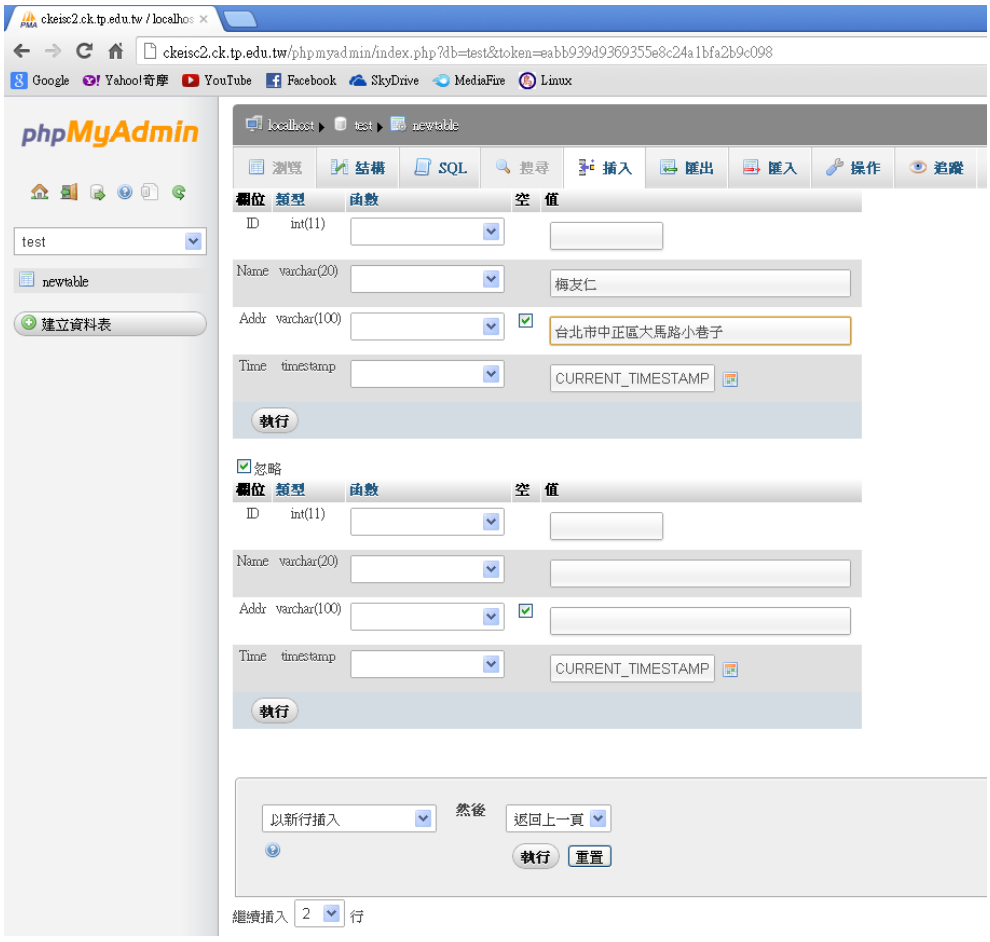

phpMyAdmin
建立資料表
test
資料庫中沒有表
建立資料表
結構
屬性
空
索引
A_I
註釋
PRIMARY

---
on update CURRENT_
---

⑤管理欄位：


phpMyAdmin
test
newtable
建立資料表
localhost test newtable
瀏覽 結構 SQL 搜尋 插入 匯出 匯入 操作 追蹤
欄位 類型 排序規則 屬性 空 預設 額外 操作
1 ID int(11) 否 無 AUTO_INCREMENT 修改 刪除 更多
2 Name varchar(20) utf8_general_ci 否 無 修改 刪除 更多
3 Addr varchar(100) utf8_general_ci 是 NULL 修改 刪除 更多
4 Time timestamp on update CURRENT_TIMESTAMP 否 CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP 修改 刪除 更多
全選 / 全不選 選中項： 瀏覽 修改 刪除 主鍵 唯一 索引 全文搜尋
列印預覽 關聯查看 規劃表結構 追蹤表
新增 1 個欄位 於表結尾 於表開頭 於 之後 ID 執行
索引:
操作 鍵名 類型 唯一 緊湊 欄位 基數 排序規則 空 註釋
編輯 刪除 PRIMARY BTREE 是 否 ID 0 A
在第 1 個欄位建立索引 執行
已用空間
類型 已用
資料 0 B
索引 1,024 B
總計 1,024 B
行統計
說明 值
格式 動態
排序規則 utf8_general_ci
行數 0
下個月 Autoindex 1
建立時間 2012 年 08 月 26 日 13:59
最後更新 2012 年 08 月 26 日 13:59

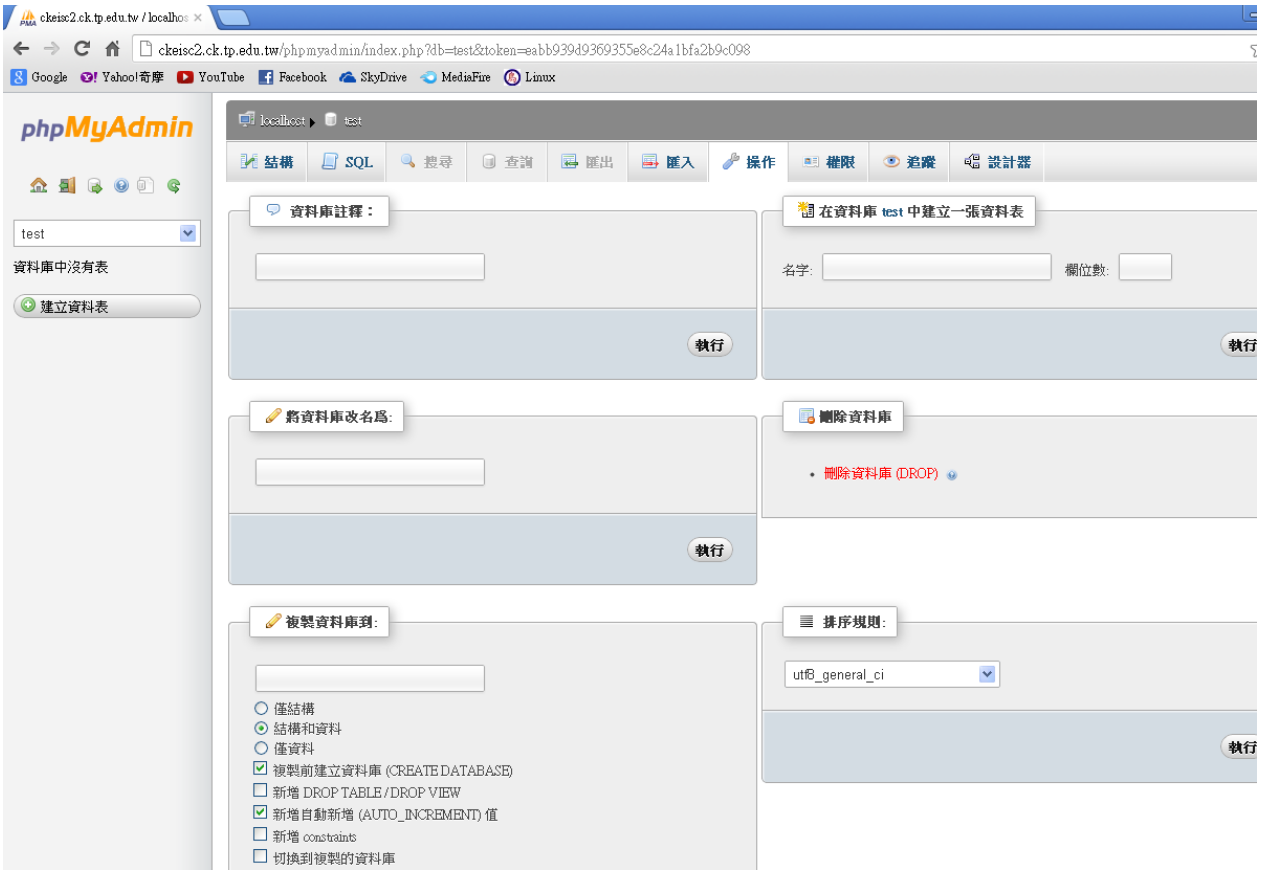
⑥新增記錄：


phpMyAdmin
test
newtable
建立資料表
localhost test newtable
瀏覽 結構 SQL 搜尋 插入 匯出 匯入 操作 追蹤
欄位 類型 函數 空 值
ID int(11)
Name varchar(20) 梅友仁
Addr varchar(100) 台北市中正區大馬路小巷子
Time timestamp CURRENT_TIMESTAMP
執行
忽略
欄位 類型 函數 空 值
ID int(11)
Name varchar(20)
Addr varchar(100)
Time timestamp CURRENT_TIMESTAMP
執行
以新行插入 然後 返回上一頁
執行 重置
繼續插入 2 行

⑦管理記錄：

⑧刪除資料表：

⑨刪除資料庫：



10 進階 PHP

10.1 插入記錄

在這個章節，我們要實作一個訊息發佈系統。這個網站包括以下六個頁面：

add.php	新增訊息頁面	連結到 index.php
login.php	會員登入頁面	連結到 index.php
index.php	首頁檢視訊息列表	傳遞 URL 參數 id 連結到 detail.php
detail.php	檢視訊息詳細資訊	傳遞 URL 參數 id 連結到 edit.php 與 delete.php
edit.php	編輯訊息頁面	連結到 index.php
delete.php	刪除訊息頁面	連結到 index.php

請先建立一個 publish 資料庫，並於其中建立一個 news 資料表，用來儲存發佈的訊息。該資料表的結構如下所示：

欄位名稱	資料型態	空值設定	其他設定
nIndex(主鍵)	int(11)	不允許 Null	auto_increment
nTitle	varchar(100)	不允許 Null	utf8_general_ci
nName	varchar(20)	不允許 Null	utf8_general_ci

nContent	text	不允許 Null	utf8_general_ci
nTime	timestamp	不允許 Null	ON UPDATE CURRENT_TIMESTAMP

另外，請在 publish 資料庫之下再建立一個 user 資料表，用來儲存會員資訊。該資料表的結構如下所示：

欄位名稱	資料型態	空值設定	其他設定
uIndex(主鍵)	int(11)	不允許 Null	auto_increment
uUsername	varchar(50)	不允許 Null	utf8_general_ci
uPassword	varchar(50)	不允許 Null	utf8_general_ci

完成這些基本設定後，便能開始進行 PHP 程式碼的撰寫了。插入記錄的的流程是：設定 MySQL 連線、處理輸入字串、執行 MySQL 指令。我們已經準備好 add.php 的表單了，以下是它的程式碼，請跟著本文的指示操作：

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html;charset=utf-8"/>
  <title>發佈最新訊息</title>
  <style type="text/css"/>
     .title_font {
        color: #FFF;
        background-color: #039;
     }
  </style>
</head>
<body>
<form action="<?php echo $FormAction; ?>" id="form1"
name="form1" method="POST">
  <table width="500" border="1" align="left" cellpadding="1"
  cellspacing="1">
     <tr>
        <td colspan="2" class="title_font"><strong>發佈最新訊息
        </strong></td>
     </tr>
     <tr>
        <td width="70" class="title_font">標題</td>
        <td width="417"><input type="text" name="Title"
        id="Title"/></td>
     </tr>
```

```
    <tr>
      <td class="title_font">發佈人</td>
      <td><input type="text" name="Name" id="Name"/></td>
    </tr>
    <tr>
      <td class="title_font">內容</td>
      <td><textarea name="Content" id="Content" cols="45"
      rows="5">
      </textarea></td>
    </tr>
    <tr>
      <td colspan="2" align="right"><input type="submit"
      name="button" id="button" value="送出"/></td>
    </tr>
  </table>
</form>
</body>
</html>
```

首先，要設定 MySQL 連線。請建立一個 publish_conn.php 的檔案，輸入連線所需的程式碼，往後若要使用這個連線，只要使用 require_once('publish_conn.php');將該檔案置入於網頁中即可。

```
<?php
//伺服器位置
$hostname = "localhost";
//資料庫名稱
$database = "publish";
//使用者帳號
$username = "root";
//使用者密碼
$password = "root";
//使用mysql_pconnect()連線到伺服器，若失敗則傳回錯誤訊息
$publish_conn = mysql_pconnect($hostname, $username, $password)
or trigger_error(mysql_error(),E_USER_ERROR);
//設定文字編碼為UTF-8
mysql_query("SET NAMES utf8");
?>
```

接著我們要處理輸入的字串，請開啟 add.php，我們將在這個網頁原始碼的最上端加入以下程式碼。

```php
<?php
//置入publish.php
require_once('publish_conn.php');
if(!function_exists("SQLValue")){
function SQLValue($theValue, $theType){
    //使用htmlentities()將特殊符號轉換為名稱實體參照
    htmlentities($theValue, ENT_QUOTES, "UTF-8");
    //若伺服器有開啟「自動為輸入的字串加上跳脫字元」功能，則用stripslashes()
    將跳脫字元去掉，這項功能在PHP6以後已經被取消
    if(PHP_VERSION<6){
        $theValue = get_magic_quotes_gpc() ?
        stripslashes($theValue) : $theValue;
    }
    //使用mysql_real_escape_string()或mysql_escape_string()為輸入的字
    串加上跳脫字元，前者功能比較完善，但舊版PHP並不支援
    $theValue = function_exists("mysql_real_escape_string") ?
    mysql_real_escape_string($theValue) :
    mysql_escape_string($theValue);
    //處理資料型態
    switch($theType){
      case "text":
        $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
        break;
      case "int":
        $theValue = ($theValue != "") ? intval($theValue) : "NULL";
        break;
      case "double":
        $theValue = ($theValue != "") ? doubleval($theValue) : "NULL";
        break;
    }
    return $theValue;
}
}
```

最後就是執行 MySQL 指令的部份了，請接續上一個步驟，將最後一段程式完成。

```php
$FormAction = $_SERVER['PHP_SELF'];
if (isset($_POST["Title"])){
    //組合SQL指令
    $insertSQL = sprintf("INSERT INTO news(nTitle, nName, nContent)
    VALUES (%s, %s, %s)",
        SQLValue($_POST['Title'], "text"),
```

```
        SQLValue($_POST['Name'], "text"),
        SQLValue($_POST['Content'], "text"));
    //連線到資料庫
    mysql_select_db($database, $publish_conn);
    //執行SQL指令，若失敗則傳回錯誤訊息
    $Result = mysql_query($insertSQL, $publish_conn)
    or die(mysql_error());
    //插入記錄後跳轉至index.php
    $insertGoTo = "index.php";
    header(sprintf("Location: %s", $insertGoTo));
}
?>
```

10.2 建立資料集(Recordset)

接下來的建立資料集就比較簡單了，我們已經準備好 index.php 的表格了，以下是它的程式碼，請跟著本文的指示操作：

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type"
    content="text/html;charset=utf-8"/>
    <title>最新訊息</title>
    <style type="text/css"/>
      .title_font {
         color: #FFF;
         background-color: #039;
      }
    </style>
</head>
<body>
    <table width="500" border="1" align="center" cellpadding="1"
    cellspacing="1">
       <tr>
           <td colspan="3" class="title_font">
           <strong>最新訊息</strong></td>
       </tr>
       <tr>
           <td width="228" class="title_font">標題</td>
           <td width="114" class="title_font">發佈人</td>
           <td width="140" class="title_font">時間</td>
```

```
        </tr>
        <tr>
            <td></td>
            <td></td>
            <td></td>
        </tr>
        <tr>
            <td colspan="3">共有筆訊息</td>
        </tr>
    </table>
</body>
</html>
```

藉由 MySQL 的 SELECT 指令取得的資料就稱為「資料集」，它是一個陣列，索引值就是資料表的欄位名稱，每執行一次 mysql_fetch_assoc()就會傳回一筆記錄。請在網頁原始碼的最上端加入以下程式碼，以建立資料集。

```
<?php
    //置入publish.php
    require_once('publish_conn.php');
    //連線到資料庫
    mysql_select_db($database, $publish_conn);
    //設定SQL指令
    $selectSQL = "SELECT * FROM news";
    //執行SQL指令，若失敗則傳回錯誤訊息
    $Result = mysql_query($selectSQL, $publish_conn)
    or die(mysql_error());
    //使用mysql_fetch_assoc()取得一筆記錄
    $rs_row = mysql_fetch_assoc($Result);
    //使用 mysql_num_rows()取得記錄筆數
    $rs_num = mysql_num_rows($Result);
?>
```

接著我們要開始使用資料集，請如下修改網頁的 HTML 程式碼。我們用 while 迴圈搭配 mysql_fetch_assoc()以取得所有的記錄。date()可以格式化任一特定的時間，第一個參數是指定格式，第二個參數是該時間點距離 1970 年 1 月 1 日 0 時 0 分 0 秒的秒數。我們用 strtotime()將 MySQL 的 timestamp 資料型態轉換成距離 1970 年 1 月 1 日 0 時 0 分 0 秒的秒數。

```
<body>
    <table width="500" border="1" align="center" cellpadding="1"
    cellspacing="1">
        <tr>
```

```
            <td colspan="3" class="title_font">
            <strong>最新訊息</strong></td>
        </tr>
        <tr>
            <td width="228" class="title_font">標題</td>
            <td width="114" class="title_font">發佈人</td>
            <td width="140" class="title_font">時間</td>
        </tr>
        <?php do { ?>
        <tr>
            <td><?php echo $rs_row['nTitle']; ?></td>
            <td><?php echo $rs_row['nName']; ?></td>
            <td><?php echo
            date('Y年d月m日',strtotime($rs_row['nTime'])); ?></td>
        </tr>
        <?php }
        while ($rs_row = mysql_fetch_assoc($Result)); ?>
        <tr>
            <td colspan="3">共有<?php echo $rs_num; ?>筆訊息</td>
        </tr>
    </table>
</body>
</html>
<?php
    //釋放資料集
    mysql_free_result($Result);
?>
```

10.3 動態連結

現在我們要做的是能檢視訊息詳細資訊的 detail.php，它是透過 URL 參數來篩選資料。請先開啟 index.php，為輸出標題的那個儲存格加上動態連結。

```
<td>
    <a href="detail.php?id=<?php echo $rs_row['nIndex']; ?>">
    <?php echo $rs_row['nTitle']; ?></a>
</td>
```

接著就能開始撰寫 detail.php 的 PHP 程式碼了，這部份和剛剛一樣都是使用 SELECT 指令建立資料集，只不過要多增加一個篩選條件而已。我們已經準備好 detail.php 的表格了，以下是它的程式碼，請跟著本文的指示操作：

```
<!DOCTYPE html>
<html>
<head>
   <meta http-equiv="Content-Type"
   content="text/html;charset=utf-8"/>
   <title>檢視詳細訊息</title>
   <style type="text/css"/>
     .title_font {
        color: #FFF;
        background-color: #039;
     }
   </style>
</head>
<body>
   <table width="500" border="1" align="center" cellpadding="1"
   cellspacing="1">
     <tr>
        <td colspan="2" class="title_font">
        <strong>檢視詳細訊息</strong></td>
     </tr>
     <tr>
        <td width="70" class="title_font">標題</td>
        <td width="417"> [編輯][刪除]</td>
     </tr>
     <tr>
        <td class="title_font">時間</td>
        <td></td>
     </tr>
     <tr>
        <td class="title_font">發佈人</td>
        <td></td>
     </tr>
     <tr>
        <td class="title_font">內容</td>
        <td></td>
     </tr>
   </table>
</body>
</html>
```

我們要用 WHERE 子句篩選指定的資料，請在網頁原始碼的最上端加入以下程式碼。

```php
<?php
   require_once('publish_conn.php');
   if(!function_exists("SQLValue")){
      function SQLValue($theValue, $theType){
         ......
      }
   }
   $selector = "-1";
   if(isset($_GET['id'])){
      $selector = $_GET['id'];
   }
   mysql_select_db($database, $publish_conn);
   $selectSQL = sprintf("SELECT * FROM news WHERE `nIndex` = %s",
      SQLValue($selector, "int"));
   $Result = mysql_query($selectSQL, $publish_conn)
   or die(mysql_error());
   $rs_row = mysql_fetch_assoc($Result);
?>
```

接著請如下修改網頁的 HTML 程式碼。

```html
<body>
   <table width="500" border="1" align="center" cellpadding="1"
   cellspacing="1">
     <tr>
        <td colspan="2" class="title_font">
        <strong>檢視詳細訊息</strong></td>
     </tr>
     <tr>
        <td width="70" class="title_font">標題</td>
        <td width="417"><?php echo $rs_row['nTitle']; ?>
         [編輯][刪除]</td>
     </tr>
     <tr>
        <td class="title_font">時間</td>
        <td><?php echo $rs_row['nTime']; ?></td>
     </tr>
     <tr>
        <td class="title_font">發佈人</td>
        <td><?php echo $rs_row['nName']; ?></td>
     </tr>
     <tr>
```

```
        <td class="title_font">內容</td>
        <td><?php echo $rs_row['nContent']; ?></td>
      </tr>
    </table>
</body>
</html>
<?php
   mysql_free_result($Result);
?>
```

10.4 更新記錄

更新記錄會同時用到 UPDATE 指令與 SELECT 指令，請先開啟 detail.php，我們要為輸出標題的那個儲存格加上動態連結。

```
<td width="417">
<?php echo $rs_row['nTitle']; ?> 
[<a href="edit.php?id=<?php echo $rs_row['nIndex']; ?>">編輯</a>]
[<a href="delete.php?id=<?php echo $rs_row['nIndex']; ?>">刪除</a>]
</td>
```

接著就能開始撰寫 edit.php 的 PHP 程式碼了，我們要先用 UPDATE 指令更新記錄，再用 SELECT 指令搭配 URL 參數篩選並建立資料集。我們已經準備好 edit.php 的表單了，以下是它的程式碼，請跟著本文的指示操作：

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html;charset=utf-8"/>
  <title>編輯訊息</title>
  <style type="text/css"/>
     .title_font {
        color: #FFF;
        background-color: #039;
     }
  </style>
</head>
<body>
<form action="<?php echo $FormAction; ?>" id="form1"
name="form1" method="POST">
  <table width="500" border="1" align="left" cellpadding="1"
```

```
  cellspacing="1">
     <tr>
        <td colspan="2" class="title_font"><strong>編輯訊息
        </strong><input type="hidden" name="Index" id="Index"/>
        </td>
     </tr>
     <tr>
        <td width="70" class="title_font">標題</td>
        <td width="417"><input type="text" name="Title"
        id="Title"/></td>
     </tr>
     <tr>
        <td class="title_font">發佈人</td>
        <td><input type="text" name="Name" id="Name"/></td>
     </tr>
     <tr>
        <td class="title_font">內容</td>
        <td><textarea name="Content" id="Content" cols="45"
        rows="5">
        </textarea></td>
     </tr>
     <tr>
        <td colspan="2" align="right"><input type="submit"
        name="button" id="button" value="更新"/></td>
     </tr>
  </table>
</form>
</body>
</html>
```

我們要為 edit.php 加入 UPDATE 指令與 SELECT 指令，並修改網頁的 HTML 程式碼，完成後如下所示。我們將主鍵值儲存在隱藏欄位「Index」裡，提供 UPDATE 指令的 WHERE 子句使用。

```
<?php
//連線及函式設定
require_once('publish_conn.php');
if(!function_exists("SQLValue")){
   function SQLValue($theValue, $theType){
       ......
   }
}
```

```php
//更新記錄
$FormAction = $_SERVER['PHP_SELF'];
if (isset($_POST["Title"])){
   $updateSQL = sprintf("UPDATE news SET nTitle=%s, nName=%s,
   nContent=%s WHERE `nIndex`=%s",
       SQLValue($_POST['Title'], "text"),
       SQLValue($_POST['Name'], "text"),
       SQLValue($_POST['Content'], "text"),
       SQLValue($_POST['Index'], "text"));
   mysql_select_db($database, $publish_conn);
   $Result = mysql_query($updateSQL, $publish_conn)
   or die(mysql_error());
   $updateGoTo = "index.php";
   header(sprintf("Location: %s", $updateGoTo));
}
//建立資料集
$selector = "-1";
if(isset($_GET['id'])){
   $selector = $_GET['id'];
}
mysql_select_db($database, $publish_conn);
$selectSQL = sprintf("SELECT * FROM news WHERE `nIndex` = %s",
   SQLValue($selector, "int"));
$Result = mysql_query($selectSQL, $publish_conn)
or die(mysql_error());
$rs_row = mysql_fetch_assoc($Result);
?>
<!DOCTYPE html>
<html>
<head>
  ......
</head>
<body>
<form action="<?php echo $FormAction; ?>" id="form1"
name="form1" method="POST">
  <table width="500" border="1" align="left" cellpadding="1"
  cellspacing="1">
     <tr>
        <td colspan="2" class="title_font"><strong>編輯訊息
        </strong><input type="hidden" name="Index" id="Index"
        value="<?php echo $rs_row['nIndex']; ?>"/></td>
     </tr>
```

```
      <tr>
         <td width="70" class="title_font">標題</td>
         <td width="417"><input type="text" name="Title"
         id="Title" value="<?php echo $rs_row['nTitle']; ?>"/>
         </td>
      </tr>
      <tr>
         <td class="title_font">發佈人</td>
         <td><input type="text" name="Name" id="Name"
         value="<?php echo $rs_row['nName']; ?>"/></td>
      </tr>
      <tr>
         <td class="title_font">內容</td>
         <td><textarea name="Content" id="Content" cols="45"
         rows="5"><?php echo $rs_row['nContent']; ?>
         </textarea></td>
      </tr>
      <tr>
         <td colspan="2" align="right"><input type="submit"
         name="button" id="button" value="更新"/></td>
      </tr>
  </table>
</form>
</body>
</html>
<?php
   mysql_free_result($Result);
?>
```

10.5 會員登入

會員登入功能其實是使用 SELECT 指令加上 WHERE 子句去查詢資料表裡是否有與輸入的帳密相同的記錄，若有，則登入成功，並使用一個 SESSION 來記錄，相對的，若查詢不到，表示帳密有誤，登入失敗。我們已經準備好 login.php 的表單了，以下是它的程式碼，請跟著本文的指示操作：

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html;charset=utf-8"/>
  <title>會員登入</title>
```

```
  <style type="text/css"/>
     .title_font {
        color: #FFF;
        background-color: #039;
     }
  </style>
</head>
<body>
<form action="<?php echo $FormAction; ?>" id="form1"
name="form1" method="POST">
  <table width="300" border="1" align="left" cellpadding="1"
  cellspacing="1">
     <tr>
        <td colspan="2" class="title_font"><strong>會員登入
        </strong></td>
     </tr>
     <tr>
        <td width="100" class="title_font">帳號</td>
        <td width="187"><input type="text" name="Username"
        id="Username"/></td>
     </tr>
     <tr>
        <td class="title_font">密碼</td>
        <td><input type="password" name="Password"
        id="Password"/></td>
     </tr>
     <tr>
        <td colspan="2" align="right"><input type="submit"
        name="button" id="button" value="登入"/></td>
     </tr>
  </table>
</form>
</body>
</html>
```

請在網頁原始碼的最上端加入以下程式碼。

```
<?php
//連線及函式設定
require_once('publish_conn.php');
if(!function_exists("SQLValue")){
   function SQLValue($theValue, $theType){
```

```
        ......
    }
}
session_start();
//查詢使用者，若記錄筆數等於一則登入成功，等於零則登入失敗
$FormAction = $_SERVER['PHP_SELF'];
if (isset($_POST['Username'])){
    mysql_select_db($database, $publish_conn);
    $selectSQL = sprintf("SELECT * FROM user
    WHERE uUsername=%s AND uPassword=%s",
        SQLValue($_POST['Username'], "text"),
        SQLValue($_POST['Password'], "text"));
    $Result = mysql_query($selectSQL, $publish_conn)
    or die(mysql_error());
    $FoundUser = mysql_num_rows($Result);
    //登入成功，更新SESSION的ID，並註冊一個SESSION，最後跳轉至index.php
    if ($FoundUser){
        if (PHP_VERSION >= 5.1)
            session_regenerate_id(true);
        else
            session_regenerate_id();
        $_SESSION['LoginUser']=$_POST['Username'];
        $LoginSuccess = "index.php";
        header(sprintf("Location: %s", $LoginSuccess));
    }
    //登入失敗，跳轉至login.php
    else{
        $LoginFailed = "login.php";
        header(sprintf("Location: %s", $LoginFailed));
    }
}
?>
```

思考：如何做出(1)會員登出功能(2)會員分級制度？(提示：使用 SESSION)

10.6 刪除記錄

刪除記錄的程式碼非常簡單，只要使用 DELETE 指令搭配 URL 參數篩選並刪除資料即可。請建立一個 delete.php 的檔案，輸入以下內容。

```
<?php
    require_once('publish_conn.php');
```

```php
    if(!function_exists("SQLValue")){
        function SQLValue($theValue, $theType){
            ......
        }
    }
    session_start();
    if((!isset($_SESSION['LoginUser'])) ||
    ($_SESSION['LoginUser'] == "")){
        $NotLogin = "login.php";
        header(sprintf("Location: %s", $NotLogin));
    }
    if((isset($_GET['id'])) && ($_GET['id'] != "")){
        $deleteSQL = sprintf("DELETE FROM news WHERE `nIndex` = %s",
            SQLValue($_GET['id'], "int"));
        mysql_select_db($database, $publish_conn);
        $Result = mysql_query($deleteSQL, $publish_conn)
        or die(mysql_error());
        $deleteGoTo = "index.php";
        header(sprintf("Location: %s", $deleteGoTo));
    }
?>
```

補充：一次處理多筆記錄
⑴將每個表單元件的名稱加上「[]」，使其轉變為陣列的形式。
⑵為表單元件加上迴圈以重複各個欄位。
⑶將 MySQL 指令也搭配迴圈執行，便可完成多筆記錄的處理。

延伸學習：
我們現在講的都是一些關於文字的處理，將文字寫入 MySQL 資料庫中管理。但是有時，我們要處理的不是文字，而是檔案，將檔案存入一個資料夾中管理，網路相簿即為一例。這時便牽扯到檔案的上傳與下載、檔案格式判斷、檔案的移動與刪除等等，這些都是值得學習的。另外，PHP 豐富的函式也是值得我們用心鑽研的。

11 Dreamweaver 應用

11.1 製做 HTML 與 CSS 文件

以上教過的 HTML 與 CSS 程式碼其實都可以輕鬆地用 Dreamweaver 完成，我們只要適時修改軟體幫我們寫的程式碼以符合自己的需求即可。

11.2 Spry 的應用

使用 Spry 可以輕鬆地完成資料集、欄位驗證、選單列、標籤面版、折疊式、可收合面版、工具提示等功能，但若你想做出與眾不同的網頁，仍需學會修改軟體幫你寫好的程式碼才行。下圖為「Spry」選單的畫面，不同版本可能會略有差異。

11.3 管理網站

Dreamweaver 提供了很多實用的工具，例如「新增與管理網站」、「建立範本與可編輯區域」、「檢查連結與孤立檔案」等，可幫助我們管理網站。

11.4 伺服器行為

第十章提到的那些 PHP 程式其實都可以用 Dreamweaver 做出來。Dreamweaver 將所有有關資料庫操作的功能都集中在「伺服器行為」選單中，只要先完成網站的定義並建立資料庫連線，即可使用這些預設的伺服器行為。不過 Dreamweaver 幫我們寫的程式碼往往不會完全符合我們的需求，因此建議大家在使用 Dreamweaver 建立程式碼後，仍要手動修改一些程式的細節，以確保 PHP 的安全性並符合我們的需求。下圖為「伺服器行為」選單的畫面，不同版本可能會略有差異。

12 架站必備常識

12.1 W3C Validation Service 檢驗 HTML 與 CSS 程式的正確性

W3C 提供了方便的線上檢測工具來檢查你的網頁是否符合 W3C 的標準。檢測 HTML 與 XHTML 文件：http://validator.w3.org/。檢測 CSS 文件：http://jigsaw.w3.org/css-validator/。通過檢測後，該站將會提供認證標示，你可以將它下載回去，放在你的頁面上。

12.2 Adobe BrowserLab 預覽網站在不同瀏覽器下的顯示情形

使用 Adobe BrowserLab 可以預覽網站在不同瀏覽器下的顯示情形，以了解你的網頁是否有相容性問題。使用這個工具前必須先登入 Adobe 帳號，若你沒有 Adobe 帳號，可以免費註冊。該站的網址為：https://browserlab.adobe.com/。

12.3 檢測網路狀態的 CMD 指令 Ping (-t)與 tracert

命令提示字元(CMD)中的 Ping 指令可以檢測一個網址是否可以連線，其指令為：ping www.example.com。Ping 指令提供了多種選項，輸入 ping /?可以查詢所有可用選項的說明。其中，最常用的是 ping -t www.example.com，這個指令可以持續對該網址做 Ping 的動作，直到無法連線至該網址為止。

tracert 指令可以追蹤連線路徑，其指令為：tracert www.example.com。同樣地，tracert 指令也提供了多種選項，輸入 tracert /?即可查詢所有可用選項的說明。執行這個指令可以讓你了解該連線是在哪個環節出了問題。

學弟妹
讓網頁設計
成為建北電資的驕傲吧！