



WORDPRESS

中文佈景主題開發指南

作者：Yuxin 最後修訂 2013.7.28

WordPress 佈景主題開發指南

WordPress Theme Development Guide

製作佈景主題也好一段時間了，而做出來的佈景也比之前好很多，所以決定再來寫一次 WordPress 佈景主題的開發指南。之前雖然寫過，但是程式碼充其量只是小兒科而已，CSS 又寫得有點差。所以，這一連串更完整一點的教學就此誕生。

這次，我們將用更清楚明瞭的方式讓你更了解 WordPress 佈景開發的每一個環節。在之前的教學中，可能有很多函式及功能並沒有多加解釋，讓大家閱讀得很辛苦，在這次指南中，希望能夠讓你有清楚的了解函式的運作及功能清楚了解。

另外，這本指南中，有一部分的代碼來自許多不同網站，我會將這些網站標於最後的引註資料中，若有缺漏，請務必從作者聯絡資料中聯絡作者，我會盡快將您補上，這本指南也是因為有你才能夠被完整的編成，非常謝謝你。

在這次的佈景主題開發指南中，把控制台佈景選項的部分先拿掉，一來是因為介紹、程式碼需許多篇幅來介紹，而寫作上又比較繁瑣，所以本份指南沒有佈景選項的教學，僅在開發上做詳盡的介紹及實作。

歡迎分享此電子書，歡迎下載。

本指南採用「CC 授權—姓名標示—非商業性」。



目錄

基礎篇

1. 開發需求
2. 開發環境 — XAMPP 使用說明
3. WordPress 佈景結構
4. 開始你的 WordPress 佈景主題
5. WordPress 佈景 style.css — 樣式表、佈景資訊宣告
6. WordPress 佈景 header.php — 頁首之製作
7. WordPress 佈景 index.php — 首頁之製作
8. WordPress 佈景 footer.php — 頁尾之製作
9. WordPress 佈景 sidebar.php — 側邊欄之製作
10. WordPress 佈景 single.php — 文章內頁之製作
11. WordPress 佈景 page.php — 頁面之製作
12. WordPress 佈景 comments.php — 迴響之製作
13. WordPress 佈景 archive.php — 彙整頁之製作
14. WordPress 佈景 search.php — 搜尋結果頁之製作
15. WordPress 佈景 searchform.php — 搜尋表單之製作
16. WordPress 佈景 404.php — 找不到頁面之製作
17. WordPress 佈景 functions.php — 佈景函式庫之製作
18. WordPress 佈景 screenshot.png — 佈景縮圖

番外篇

1. WordPress Breadcrumb 麵包屑製作
2. WordPress 作者介紹使用函式實作
3. WordPress 佈景文章縮圖功能之製作
4. WordPress Og:Meta 使用函式實作
5. WordPress 在文章 more 標籤後插入廣告代碼
6. 後記

其他

1. 關於作者
2. 引註連結

基礎篇

開發需求

開發環境

一台可以正常使用的電腦，可以上網，可以裝文字編輯軟體，Windows or Mac 或其他 (Ubuntu ...)。強烈建議安裝 XAMPP or Appserv 於本機端，以方便查看目前開發的情況（於下一章節說明）。

筆者的開發環境是 Windows 7。

開發者成就

你需要擁有基礎 HTML 與 CSS 的成就。最好，也要有基礎 PHP 的成就，只要懂 function, if-else, variable, include 大概就可以進行這一連串 WordPress 的任務了。

- HTML
- CSS
- PHP

開發者資源

想學 HTML, CSS, PHP 的讀者可至 W3School 閱讀相關文件學習([簡體中文](#))([英文](#))，或者自行購書學習。

推薦開發軟體

推薦 Sublime Text 2 來進行開發。關於 Sublime Text 2 的詳細介紹以及用法，可參考以下網站：[Sublime Text 手冊](#)。

請不要忘記 Sublime Text 2 強大的 Package 功能，並強烈推薦安裝 Emmet 快速 Coding 套件。

你可能會喜歡的框架 (Framework)

1. [Twitter Bootstrap](#)
2. [Foundation](#)
3. [Gumby](#)

開發環境 — XAMPP 使用說明

在佈景開發過程中，我們可以有兩種方式：一是將 WordPress 裝在虛擬主機中，然後製作佈景的時候會一直不斷將檔案使用 FTP 上傳至虛擬主機；二是直接在自己的電腦中裝設 WordPress，這樣不但不需要每次都使用 FTP 上傳，而且只要打開瀏覽器輸入 <http://localhost/> 就可以看到目前的開發狀態。之後開發完成後，再統一上傳到自己的虛擬主機就可以了。

沒錯，我們這次要使用的就是將 WordPress 安裝在自己的電腦，在自己的電腦中建立佈景主題，進行 WordPress 佈景主題世紀終極大開發。

我們要利用的工具是 XAMPP。XAMPP 是一個結合 Apache、PHP、PERL、MySQL 的軟體，相當方便。

XAMPP 基本介紹

正體中文首頁：http://www.apachefriends.org/zh_tw/index.html

分類下載頁面：http://www.apachefriends.org/zh_tw/xampp.html




Windows 下載頁：http://www.apachefriends.org/zh_tw/xampp-windows.html

Mac OSX 下載頁：http://www.apachefriends.org/zh_tw/xampp-macosx.html

Linux 下載頁：http://www.apachefriends.org/zh_tw/xampp-linux.html

Windows 安裝說明

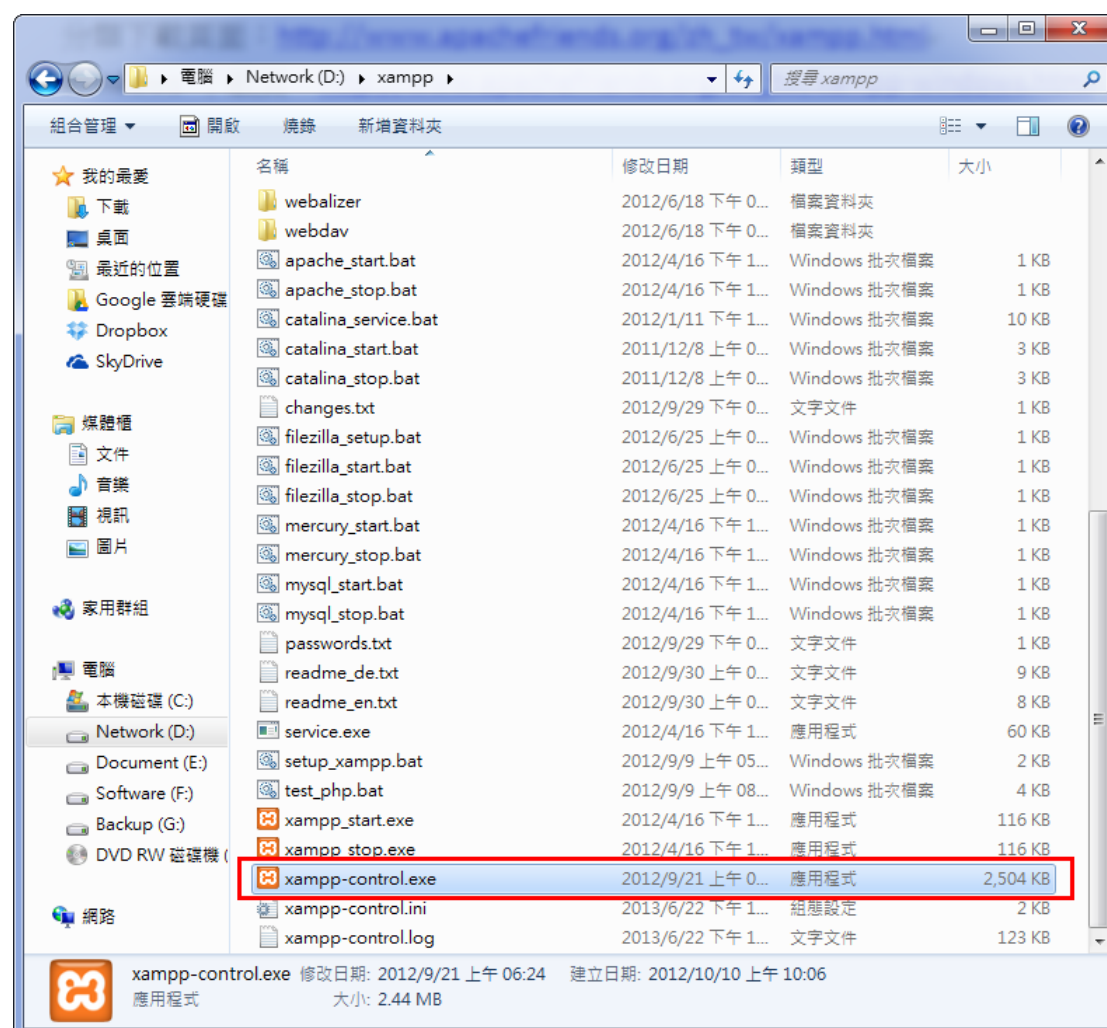
進到 [Windows 下載頁](#)。如下圖，你可以點擊「自動安裝程式」、「ZIP」或「7zip」進行 XAMPP 的下載。如果你對免安裝程式不熟悉，推薦你使用「自動安裝程式」版本；若您熟悉，推薦選擇「7zip」版本，下載快且不留痕跡。

XAMPP Windows 1.8.1版, 30.9.2012		
版本	大小	內容
XAMPP Windows 1.8.1		
Apache 2.4.2, MySQL 5.5.27, PHP 5.4.7, OpenSSL 1.0.1c, phpMyAdmin 3.5.2.2, XAMPP Control Panel 3.1.0, Webalizer 2.23-04, Mercury Mail Transport System v4.62, FileZilla FTP Server 0.9.41, Tomcat 7.0.30 (with mod_proxy_ajp as connector), Strawberry Perl 5.16.0.1 Portable For Windows 2000, XP, Vista, 7.		
 自動安裝程式	99 MB	自動安裝程式 MD5 checksum: 2c067c31725fda3c71c6d43483b4df4c
 ZIP	184 MB	ZIP 壓縮檔 MD5 checksum: 924e9cdc0fc49984e0c4916aa8f31c18
 7zip	84 MB	7zip 壓縮檔 MD5 checksum: 462f6bc3c9e96a8c9228927ff8e0d217

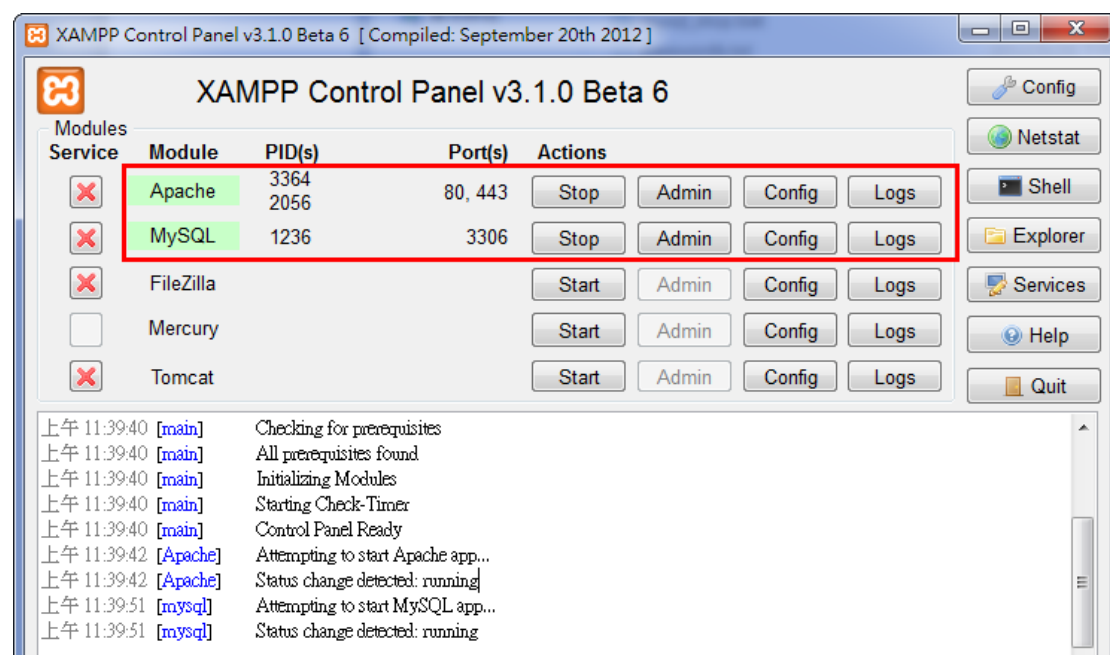
打開你安裝 XAMPP 的目錄。往下拉，你會看到一個「xampp-control.exe」，將它打開，你會看到管理各項服務開啟的介面。

注意：如果你使用的是 ZIP 或 7-zip 版本，在開啟 xampp-control.exe 之前請先

執行 `setup_xampp.bat`，再開啟 `xampp-control.exe`，才能正常開啟各項服務。



開啟 `xampp-control.exe` 之後，要安裝 WordPress，我們必須將 Apache 及 MySQL 這兩項服務開啟，只要按一下 Start 即可。如下圖，為開啟狀態。

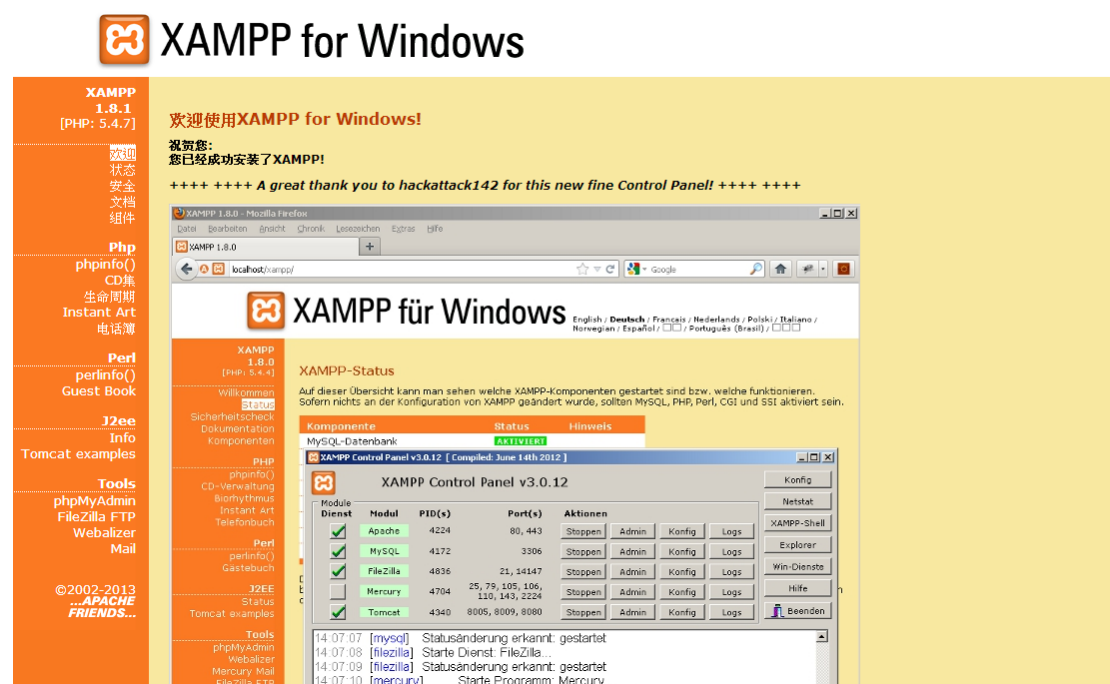


注意：某些程式可能會擋住 Apache 使用的 80port，如 Skype，若發生請關掉占用 80port 的程式或修改其設定。

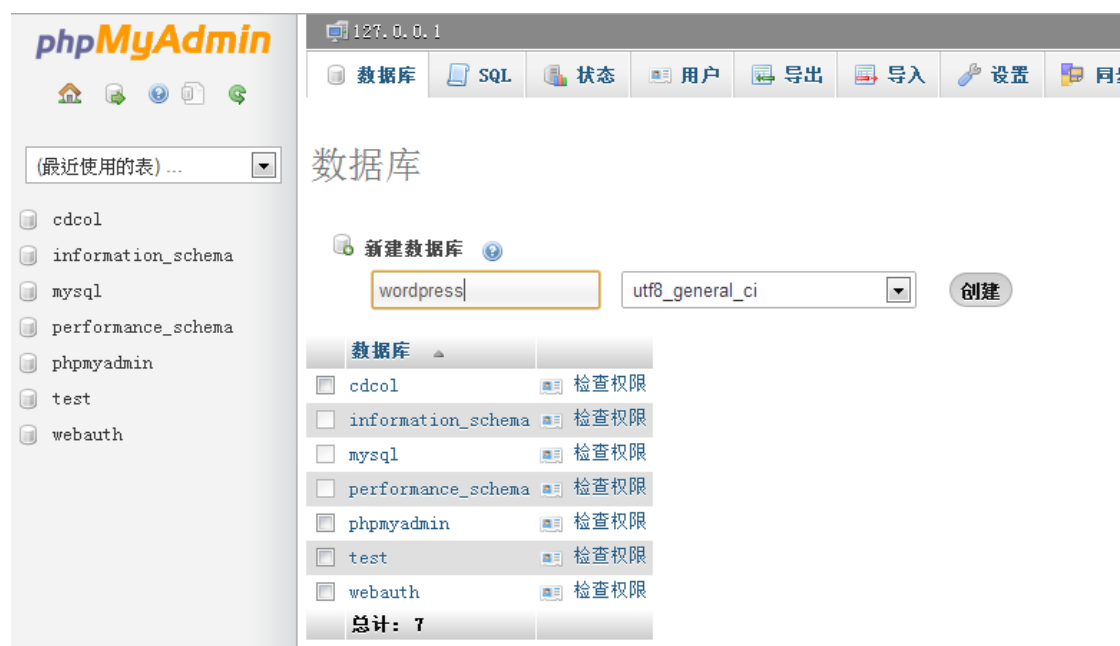
XAMPP 網站目錄

XAMPP 存放網站檔案的地方在「htdocs」資料夾。

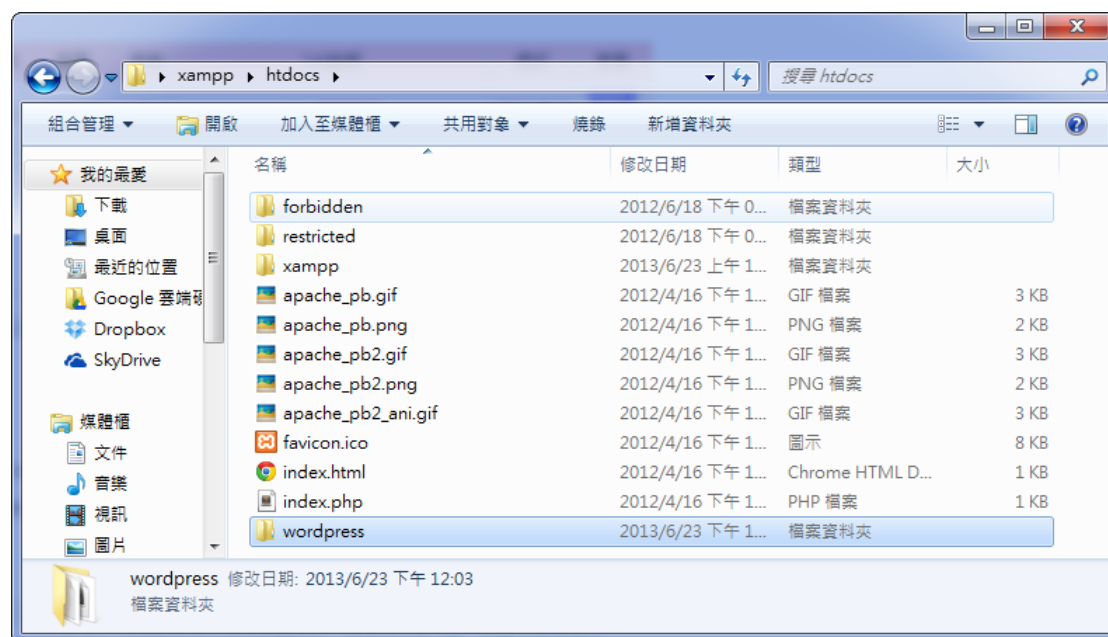
你現在可以打開瀏覽器，並且輸入「<http://localhost>」或「<http://127.0.0.1>」，你講會看到以下頁面：



現在，你可以看一下關於 XAMPP 的各項資訊。或者，你可以打開 htdocs 目錄，開始安裝 WordPress。安裝步驟與一般安裝方式一樣，你需要先打開 phpMyAdmin (<http://localhost/phpmyadmin>)，並新增資料庫。如下圖：



接著，你需要到 WordPress 官網下載最新的 WordPress。下載完後，將完整檔案放至 htdocs 資料夾內。如下圖：



接著，打開 <http://localhost/wordpress/> 即可開始安裝。另外，安裝過程中的資料庫帳號及密碼，帳號預設為 root，密碼預設為空。（有修改請填你修改後的）。資料庫名稱請填上一張圖片中新建數據庫你填的資料庫名稱。

最後安裝成功後會進到 WordPress 的控制台。如果你能夠進去，這樣初始的本機開發環境就算完成了，恭喜完成成就十分之一。

WordPress 佈景結構

在開始 Coding 之前，我們要先來了解 WordPress 整個佈景的結構。古人言：「工欲善其事，必先利其器」，而基本的 WordPress 佈景結構知識就是我們的器。

本份教學佈景主題檔案

如果你想要直接下載這一個佈景主題的檔案回去慢慢研究或者要除錯，你可以下載我範例中所有的檔案。同時，你也可以看範例最後做出來的網站。如下連結：

- [WordPress 佈景主題開發指南](#)

WordPress 佈景主題目錄位置

WordPress 的佈景主題是放在 `/wp-content/themes` 裡，另外在 `themes` 資料夾裡又分很多資料夾表示不同佈景。

WordPress 佈景主題檔案

通常，一個完整的 WordPress 佈景主題有以下檔案：

- `style.css`
- `header.php`
- `footer.php`
- `index.php`
- `single.php`
- `page.php`
- `sidebar.php`
- `comments.php`
- `archives.php`
- `search.php`
- `searchform.php`
- `404.php`
- `functions.php`

而這些檔案在 `/wp-content/themes/佈景名稱/` 裡面。

WordPress 佈景主題資訊

WordPress 佈景主題的等等資訊，都寫在 `style.css` 裡頭（同時，`style.css` 也是樣式表），每個佈景主題一定需要這個檔案。

開始你的 WordPress 佈景主題

首先，請在 `/wp-content/themes` 裡新增一個資料夾，請使用英文小寫名稱（可使用阿拉伯數字），且不要有任何標點符號。

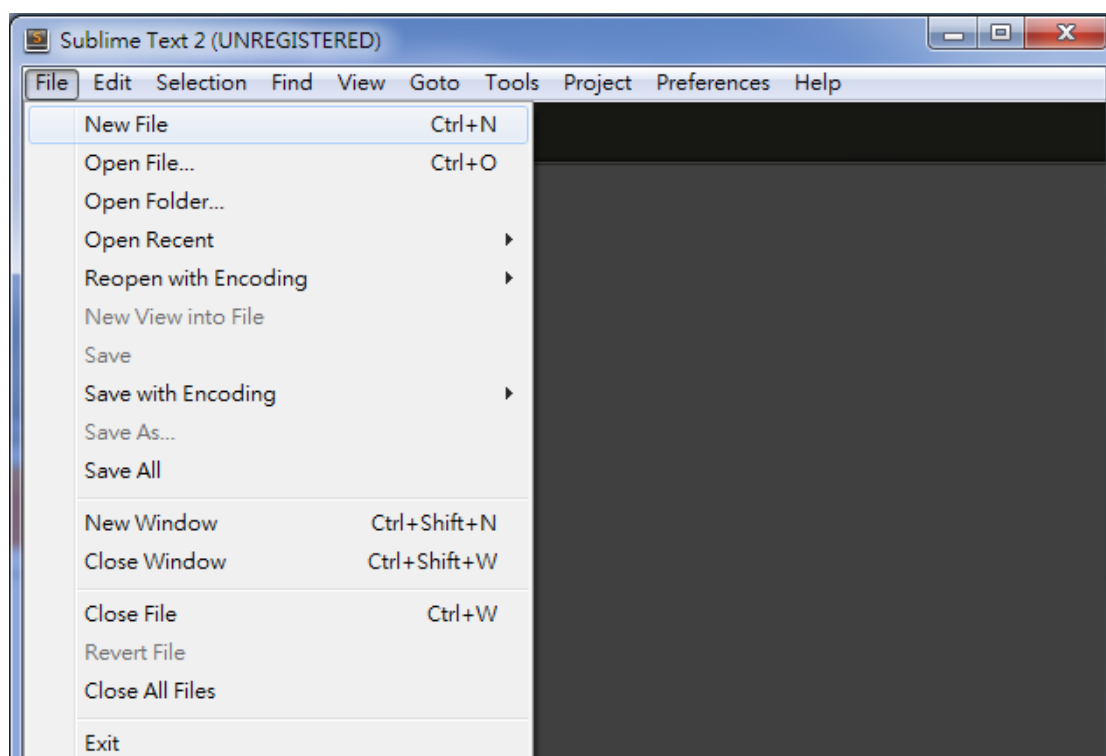
接著，在你剛剛新增的資料夾裡頭，新增以下檔案，檔案內容會是空白的，並請注意副檔名是否正確、全為小寫。

- `style.css`
- `header.php`
- `footer.php`
- `index.php`
- `single.php`
- `page.php`
- `sidebar.php`
- `comments.php`
- `archives.php`
- `search.php`
- `searchform.php`
- `404.php`
- `functions.php`

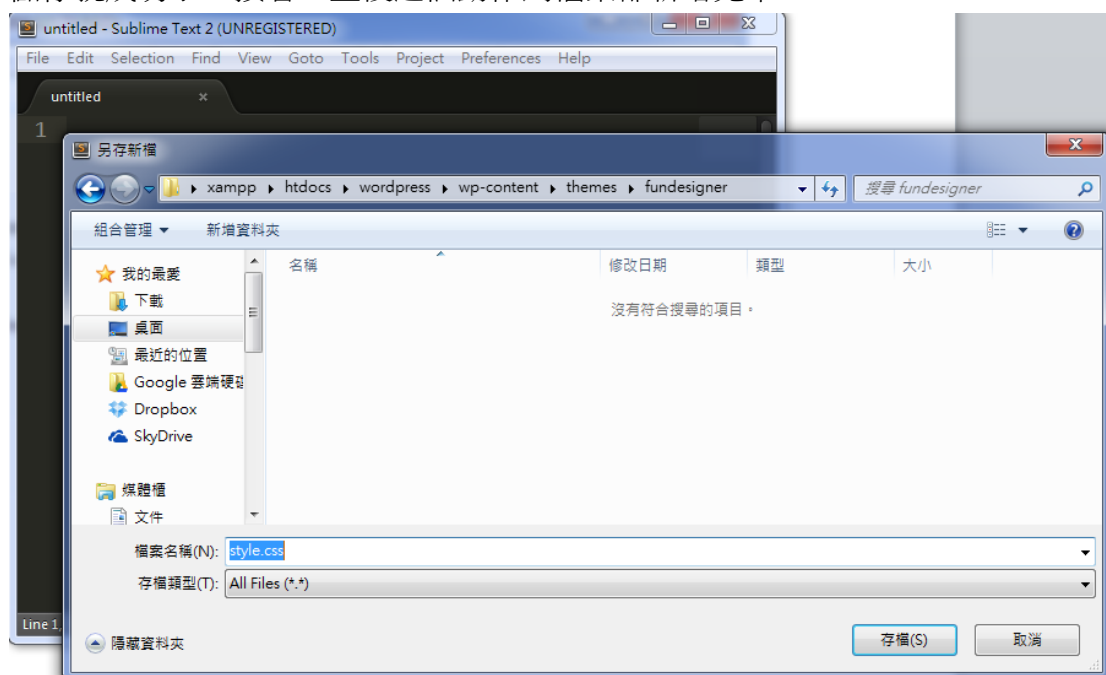
雖然新增的都是 PHP 檔案，但是我們會將 HTML 和 PHP 混合在一起寫。另外，請特別注意 PHP 語法之 `<?php` 與 `?>` 之開啟與關閉碼，以免發生錯誤。

(Q&A) 怎麼新增 PHP 及 CSS 檔案？

以 Sublime Text 2 為例，點左上角的 File，再按一下 New File（快速鍵：Ctrl+N）。



接著，按 `Ctrl+S` 另存新檔，檔案名稱填入上面列表中的檔案名稱。接著按一下儲存就成功了。接著，重複這個動作到檔案都新增完畢。



恭喜，現在你可以準備好開始 Coding 了，請準備好你的必勝布條，戴在頭上。一切都準備好了嗎？開始衝！進到下一章節吧。

WordPress 佈景 style.css

在 style.css 中，我們將設定 WordPress 控制台將擷取的佈景主題各項資訊，如名稱、說明、作者、版號等等。

打開 style.css。

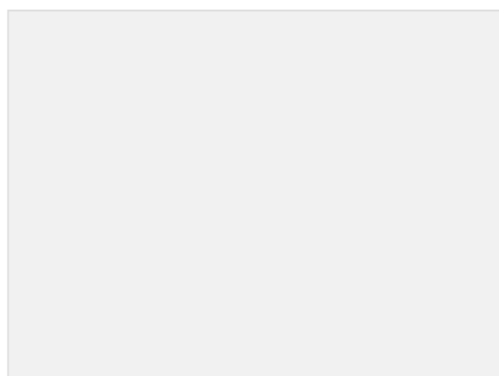
WordPress 佈景主題資訊

首先，在 style.css 中，加入以下註解，並加以修改：

```
/*
Theme Name: Fundesigner
Theme URI: http://fundesigner.net
Description: A Fundesigner Theme.
Author: Yuxin
Author URI: http://fundesigner.net
Version: 1.0
Tags: white, two-columns
License: GNU General Public License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
*/
```

這樣，在控制台的地方你就會看到你所設定的新佈景主題。如下圖的左方：

可用佈景主題

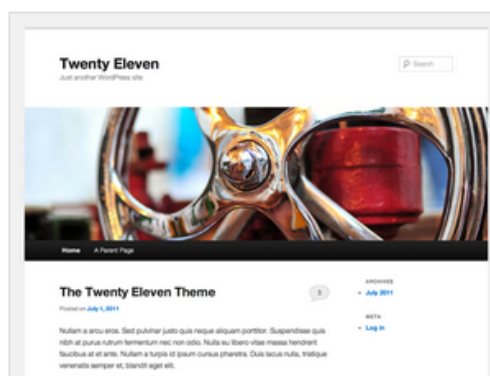


Fundesigner

作者: [Yuxin](#)

[啟用](#) | [即時預覽](#) | [細節](#)

[刪除](#)



貳零壹壹

作者: [WordPress 團隊](#)

[啟用](#) | [即時預覽](#) | [細節](#)

[刪除](#)

現在可以按啟用。當然，現在啟用網站會是一片空白，我們還沒有做頁面。不過，先來看一下上面的每一行的用意：

- Theme Name: 佈景主題名稱
- Theme URI: 佈景主題網址
- Description: 佈景主題說明
- Author: 佈景主題作者
- Author URI: 佈景主題作者網站
- Version: 佈景主題版本號
- Tags: 佈景主題標籤
- License: 佈景主題授權
- License URI: 佈景主題授權書

Style.css Reset

一般來說，在開發網站要撰寫 CSS 之前，我們都會先利用將各項元素 reset 掉，以避免各瀏覽器對各元素的初始設定不同。

所以，請在 style.css 內（註解的後面）加入以下 reset CSS 的樣式碼：

```
@import
url(http://fonts.googleapis.com/css?family=Open+Sans:
300,400);

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
```

```
    font: inherit;
    vertical-align: baseline;
}
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
}
blockquote, q {
    quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: '';
    content: none;
}
table {
    border-collapse: collapse;
    border-spacing: 0;
}
```

另外，第一行的 `@import` 純粹是因為在後面我們會使用到 Open Sans 字體，先把它引入進來。

加入一些基本元素樣式

在上面的樣式碼中，我們將全部的元素預設樣式都清掉了，所以我們現在就可以來套上自己希望元素長怎樣的樣式了。如 `a`, `ul`, `blockquote` 等等。將下面的樣式碼加入 `style.css`（緊接著上面的最後）。

```
a {
    color: #22A2EB;
    text-decoration: none;
}
```



```
a:hover {
    text-decoration: underline;
}
img {
    max-width: 100%;
    height: auto;
}
ul {
    margin-left: 25px;
}
li {
    list-style: square;
    margin: 5px 0;
    color: #555;
}
h1 {
    font-size: 32px;
    color: #555;
    line-height: 1.8;
    font-weight: 300;
}
h2 {
    font-size: 26px;
    line-height: 1.48;
}
h3 {
    font-size: 20px;
    line-height: 1.48;
    color: #777;
}
```

接下來的開發

接下來，只要後面的章節提到 CSS 樣式碼，都是寫在 style.css 中。(寫在上面那段的後面接著寫。)。另外，請注意括號的配對，不要漏掉唷。

WordPress 佈景 header.php

在 header.php 中，我們將撰寫要在網頁頂部顯示的介面。此外，header.php 也是載入 CSS、Javascript 的地方，以及各項 meta 標籤。

<head> 頭部標籤同樣會放在這個檔案裡。

網頁基本元素 — <html>, <head>, <body> 等

首先，我們先將最基本的元素加到 header.php 中。其他載入 CSS, JS, Meta 則稍後再寫。我們不是神，我們要有順序有結構地寫。請將下面代碼放入空無一物的 header.php：

```
<!DOCTYPE html>
<html <?php language_attributes(); ?>>
  <head>
    <meta charset="<?php bloginfo('charset');?>" />
    <title><?php
      if (is_home()) {
        bloginfo('name');
        echo ' - ';
        bloginfo('description');
      } else {
        wp_title(' - ', true, 'right');
        bloginfo('name');
      } ?></title>
  </head>
  <body>
```

從上面的程式碼中，你可能會看到幾個你不認識的函式：

- language_attributes() — 取得網站的語系。如 zh_tw。
- bloginfo() — 網站各項重要資料，如填入 'charset' 則輸出網站編碼，填入 'url' 則輸出網站的網址，'name' 則輸出網站名稱，'description' 則輸出網站描述。官方詳細參數：[WP Codex – bloginfo\(\)](#)
- wp_title() — 網站當前頁面名稱。官方詳細參數：[WP Codex – wp_title\(\)](#)

<head> 內標籤 — <meta>, <link>

接著，我們要開始加入跟佈景有相關性的標籤了。另外，你也可以加入自己需要

的，如 jQuery Plugin 等等。在這裡僅加入較基本的幾項標籤。請加在 `</title>` 之後。

```
<?php wp_head(); ?>
    <meta name="viewport"
content="width=device-width, initial-scale=1" />
    <link rel="pingback" href="<?php
bloginfo( 'pingback_url' ); ?>" />
    <link href="<?php
bloginfo('template_directory') ?>/style.css"
media="screen" rel="stylesheet" type="text/css" />
```

新函式說明：

- `wp_head()` — WordPress 頭部的 Hook。某些外掛會利用此處將外掛需要的 Javascript 檔案載入，我們也可以利用此處載入需要載入的檔案。
- `bloginfo('pingback_url')` — 網站 Pingback 的網址。
- `bloginfo('template_directory')` — 網站套用中佈景之路徑，如「`http://網址/wp-content/themes/fundesigner`」

網站頂部介面 — 選單、Logo 及其他

接著，我們要開始來寫會真正顯示在網站上的東西了！是不是很期待呢？其實也不用太期待，接下來的東西，並不只是複製貼上就完成了，除了複製貼上之外，你還要自己針對自己想要的樣式下去做修改、調整，讓佈景有你的風格，是你想要見到的。

而我們將在 `header.php` 中製作選單、Logo 及整體介面上半部。

首先，在 `header.php` 的 `<body>` 後面加入以下代碼：

```
<div class="container">
    <header class="header">
        <h1 class="title"><a href="<?php
bloginfo('url'); ?>">樂在設計</a></h1>
        <nav>
            <?php
wp_nav_menu( array( 'theme_location' =>
'primary-menu' ) ); ?>
        </nav>
    </header>
```

為什麼我現在去開網站沒有任何東西？因為現在 WordPress 讀取的是 index.php，而我們還沒有在 index.php 內將 header.php 嵌入進來，所以還沒有東西。不過先別急，我們先把這部分完成，到了 index.php 你就能一覽廬山真面目了。

等等！我們還沒有介紹 wp_nav_menu() 這個函式。這個函式說穿了就是把在控制台→外觀→選單中所設定的選單叫出來。'theme_location' 則是告訴系統要的是哪一個選單。因為我們還沒有定義出 'primary-menu' 這個選單，所以我們打開 functions.php 並加入以下函式：

```
<?php
register_nav_menus(
    array(
        'primary-menu' => __( '主選單' )
    )
);
?>
```

這樣，你在控制台→外觀→選單中應該就會看到能夠設定的選項，如下圖，而你可以新增幾個項目進去：（注意佈景主題位置→主選單必須被設定。）



再來，我們要為之前的 HTML 標籤做 CSS 樣式美化。將以下的 CSS 樣式碼加入 style.css：

```
/* Main */
body {
    background: #FAFAFA;
    font-family: "Open Sans", sans-serif;
```

```
}  
.container {  
    width: 1000px;  
    margin: 0 auto;  
}  
.header {  
    width: 100%;  
    text-align: center;  
}  
.title {  
    margin: 60px 0;  
}  
.title a {  
    font-size: 60px;  
    color: #666;  
    text-decoration: none;  
}  
.menu li {  
    display: inline-block;  
    padding: 5px 10px;  
}  
.menu li a {  
    background: #FFF;  
    color: #888;  
    padding: 8px 15px;  
    font-family: "Open Sans", sans-serif;  
    font-weight: 300;  
    text-decoration: none;  
}  
.menu li a:hover {  
    background: #AAA;  
    color: #FFF  
}
```

這樣，header.php 就算完成囉！請接著下一章 index.php 看，就能夠看到 header.php 的效果了！

WordPress 佈景 index.php

現在，我們要進行第一個重頭戲。那就是首頁（index.php）的製作。首頁是一個網頁給人的第一印象，所以必須要做得好，訪客留下來閱讀的機會也會增高。而在 index.php 中，我們會將 header.php, footer.php, sidebar.php 檔案以函式的方式載入。

而這次，我們只做最基本的讓網站文章跑出來，並做出一個好看的頁碼讓訪客能夠進到下一頁繼續看文章。

基本格局

首先，我們先把基本格局寫好，把 header.php, footer.php, sidebar.php 分別使用 get_header(), get_footer(), get_sidebar() 函式載入到 index.php。

```
<?php get_header(); ?>
<div class="content">
  <div class="article">

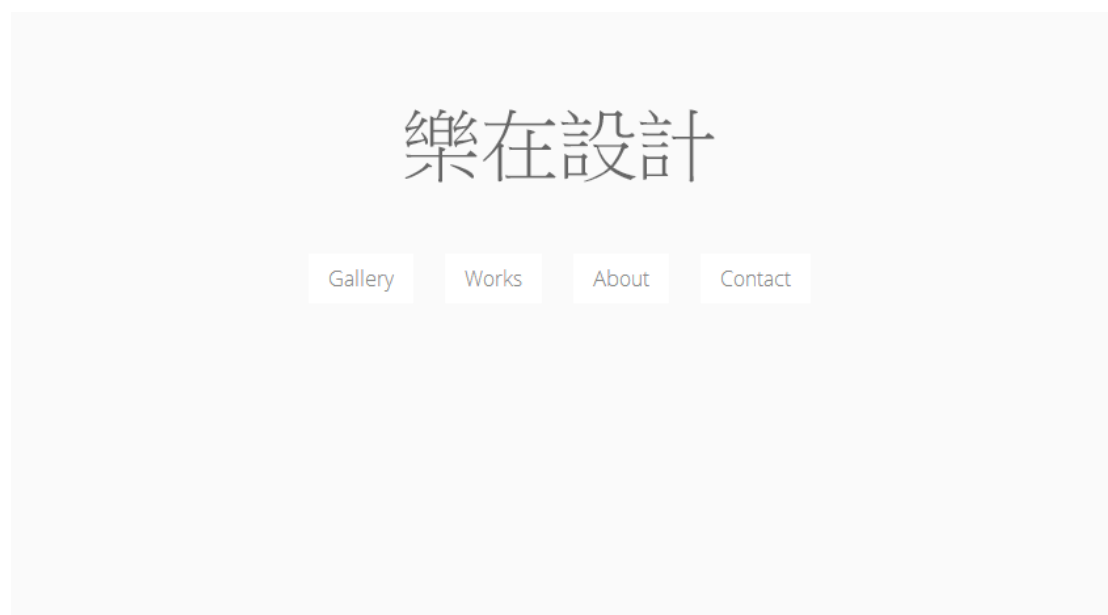
    </div>
    <div class="sidebar">
      <?php get_sidebar(); ?>
    </div>
  </div>
  <?php get_footer(); ?>
```

接下來，我們為上面的 HTML 標籤加上 CSS，讓網站能夠呈現兩欄的模樣，在 style.css 加入以下樣式碼：

```
/* Main Column */
.content {
  float: left;
  width: 100%;
  margin-top: 30px;
}
.article {
  width: 70%;
  float: left;
}
```

```
.sidebar {
    width: 30%;
    float: left;
}
```

現在，你可以打開 <http://localhost> 看看目前網站長得怎樣了！目前的樣子，大部分是在上一節 `header.php` 所做出來的樣式。如下圖：



而，如果你想要修改樣式，你可以回到 `header.php` 及 `style.css` 進行修改！

迴圈輸出網站文章

在上一步的代碼中，`.article` 是顯示文章的區塊，而 `.sidebar` 則是顯示側邊欄的區塊。現在，因為是首頁，所以我們要在 `.article` 裡面一次跑出好幾篇文章，到一定數量即換頁。所以，把下面的代碼放到 `.article` 裡面，接著再看詳解。

```
<?php while ( have_posts() ) : the_post(); ?>
    <article class="article-content">
        <h1 class="article-title"><a href="<?php
the_permalink(); ?>"><?php the_title(); ?></a></h1>
        <div class="article-meta">
            <span><?php the_time('n / j,
Y'); ?></span><span> / </span>
            <span><?php the_category(' ,
'); ?></span><span> / </span>
            <span><?php the_tags(' , ' , ' ,
```

```
''); ?></span>
    </div>
    <?php the_content(); ?>
    <div class="clearfix"></div>
  </article>
<?php endwhile; ?>
```

在上面的代碼中，我們用了好幾個還沒出現過的函式，以下是解釋：

- `while (have_posts())` — `have_posts()` 為判斷是否在資料庫中有文章，若有則回傳 `true`，並進入 `while` 迴圈中。
- `the_post()` — 載入文章相關函式，如需使用下面幾個跟文章資料有關的函式必須先加入 `the_post()` 函式，才會正常。
- `the_title()` — 輸出文章標題。
- `the_time(' n / j, Y')` — 輸出文章發表的時間，參數為顯示時間的格式。`n / j, Y` 表示格式為 月 / 日, 年，詳細輸出格式可參考：[WP Codex — Format Date & Time](#)
- `the_category(',')` — 輸出文章所在分類。參數為分類與分類間的分隔符號。如 分類一, 分類二。
- `the_tags(' ', ', ', '')` — 輸出文章之標籤。第一個參數為在輸出標籤前的文字，假設為 '標籤：' 則整個函式會輸出「標籤：標籤 1, 標籤 2」。第二個參數為標籤與標籤之間的分隔符號。第三個參數為在輸出標籤後的文字，假設為 '
' 則輸出文字後將會直接換行。
- `the_content()` — 輸出文章內容。
- `the_permalink()` — 輸出文章網址

接下來，我們為上面那些標籤加上稍微好看又清爽的 CSS 樣式：

```
/* Posts */
.article-content {
  margin: 30px;
  padding: 20px;
  background: #FFF;
}
.article-content p {
  font: 15px/1.7 "Open Sans", sans-serif;
  color: #565656;
  margin: 15px 0;
}
.article-title {
  font-size: 32px;
```



```
        color: #555;
        line-height: 1.8;
        font-weight: 300;
    }
    .article-meta span {
        color: #888;
        font-size: 13px;
        margin-right: 7px;
    }
    .article-meta span a{
        color: #888;
        text-decoration: none;
    }
    .article-meta span a:hover {
        text-decoration: underline;
    }
    .clearfix {
        content: ".";
        display: block;
        height: 0;
        clear: both;
        visibility: hidden;
    }
```

截至目前，佈景主題長成這樣：



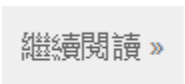
繼續閱讀字樣美化

現在，我覺得繼續閱讀的文字和其他文字除了顏色以外沒有差異性，所以我想要來做一點樣式美化它：

你會發現，在之前寫的代碼中完全沒有繼續閱讀這四個字。沒錯，因為繼續閱讀是在 `the_content()` 函式中被顯示的，所以你不曾看到他在你寫的代碼中。不過，利用 Chrome 的 Developer Tools 的檢查元素之後，我們會發現繼續閱讀那四個字的 CSS class 是 `more-link`。所以，我們就幫繼續閱讀那四個字做美化吧。

```
.more-link {  
  float: right;  
  color: #666;  
  padding: 7px 10px;  
  background: #EFEFEF;  
  text-decoration: none;  
}
```

```
.more-link:hover {
    background: #888;
    color: #FFF;
}
```



另外，上面所提到的檢查元素，只要你在 Chrome 中，將滑鼠游標移至繼續閱讀四個字上，然後按右鍵，點檢查元素，就會看到那一段的原始碼了！

精緻頁碼製作

接著，我們要為文章列表加上一個有點精緻的簡單頁碼了，讓訪客能夠換下一頁繼續閱讀。而頁碼我們用數字的方式呈現，另外還需在 functions.php 內加入一個輸出頁碼的函式。所以，我們先在 functions.php 內加入頁碼函式：

```
function wp_pagenavi() {
    global $wp_query;
    $max = $wp_query->max_num_pages;
    if ( !$current = get_query_var('paged') ) $current = 1;
    $args['base'] = str_replace(999999999, '%#%',
get_pagenum_link(999999999));
    $args['total'] = $max;
    $args['current'] = $current;
    $args['prev_text'] = '<';
    $args['next_text'] = '>';
    if ( $max > 1 ){ $pages = '<div
class="wp-pagenavi"><span class="pages">共 ' . $max . '
頁</span>';
        echo $pages . paginate_links($args) . '</div>';}
}
```

我們做出頁碼的方法是利用 WordPress 內建的 `paginate_links()` 函式，來跑出頁碼。而由於 `paginate_links()` 函式預設的顯示方法不合我們的需求，所以我們要

利用一些參數讓顯示能夠達到我們要的效果。([WP Codex paginate links](#))

- `$wp_query` — 用意是在之後的 `$wp_query->max_num_page` 用來取得總頁數，因為我們是寫在函式裡面，所以必須要把 `$wp_query` 令為全域變數。

- `get_query_var('paged')` — 取得目前所在的頁數。

接著，有一個 `$arg` 的陣列，裡面存放的資料會在下面的 `paginate_links($args)` 當作參數傳給系統。

- `$args['base']` — 用來避免第一頁頁碼會沒有連結的問題，這樣可以確保每一個頁碼都有連結可以點。(除了目前所在頁面的頁數)。

- `$args['total']` — 總頁數。

- `$args['current']` — 目前所在頁數

- `$args['prev_text']` — 上一頁顯示之文字

- `$args['next_text']` — 下一頁顯示之文字

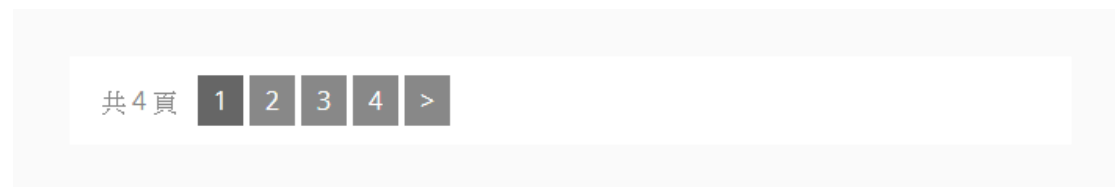
接著，來加入一些讓頁碼更好看的 CSS 樣式囉：

```
/* PageNavi */
.wp-pagenavi {
    margin: 30px;
    padding: 20px;
    background: #FFF;
}
.pages {
    color: #888;
    margin-right: 13px;
}
.page-numbers {
    padding: 5px 10px;
    color: #FFF;
    background: #888;
    text-decoration: none;
}
.page-numbers:hover {
    color: #FFF;
    background: #666;
}
.page-numbers.current {
    color: #FFF;
    background: #666;
}
```

接著，你就可以在 index.php 的 <div class="sidebar"> 之前加入呼叫函式：

```
<?php wp_pagenavi(); ?>
```

這樣，就可以做出好看又有點精緻的頁碼囉。做好之後如下圖：



WordPress 佈景 footer.php

footer.php 這一個檔案通常是拿來放一些版權資訊等等不屬於主要內容的資訊。而這個檔案同樣要把整個 HTML 標籤還尚未關閉的做一個適當的關閉，如 .container，要在 footer.php 將它關閉。

基本格式 — 將未關閉的標籤關閉

在 header.php 中，有些標籤是還沒有關閉的。而這些標籤我們要在 footer.php 將它關閉，這樣才算一份語法正確又好的網頁。在 footer.php 加入以下代碼：

```
<footer class="footer">
</footer>
</div>
<?php wp_footer(); ?>
</body>
</html>
```

- bloginfo() — 參數若為 url 則輸出網站網址；若為 name 則輸出網站名稱。
- wp_footer() — 若有登入，在網站中每個頁面的最上面都會有一條黑色的控制條。此函式即為載入該條之函式。

最後，我們將 body 與 html 一同關閉。

加入版權資訊

另外，在網頁底部幾乎每個網站都會放置版權資訊，有時候也有放聯絡資料。而在這邊我們會在 footer.php 放上網站的版權資訊。在 footer.php 的 <footer class="footer"> 後加上：

```
<div class="copyright">Copyright © 2013 <a href="<?php
bloginfo("url"); ?>"><?php
bloginfo("name"); ?></a></div>
```

接著，為這些 HTML 標籤加上一些 CSS 吧！

```
/* Footer */
.footer {
    margin: 0px 30px 15px 30px;
    float: left;
    padding: 20px;
}
.copyright {
    color: #888;
    font-size: 13px;
}
.copyright a {
    color: #666;
    text-decoration: none;
}
.copyright a:hover {
    color: #666;
    text-decoration: underline;
}
```

最後，打開瀏覽器 localhost，你會看到 footer 乖乖的在網站的最底部！

Copyright © 2012-2013 樂在設計

WordPress 佈景 sidebar.php

在 sidebar.php 中，我們可以放置一些如相關文章、搜尋等等模組讓訪客有需求時去使用。而 sidebar.php 的寫法非常簡單，甚至在 sidebar.php 中只要三行。不過，我們得另外在 functions.php 中先去註冊模組，這樣你就可以在控制台的外觀 > 模組中去新增模組，之後在 sidebar.php 中被顯示出來。

註冊模組

如果你現在去控制台的外觀裡面，你會看不到模組選單。是的，因為我們要先告訴系統我們要使用這個功能，也就是註冊一個模組。所以，請將下面的代碼加入 functions.php。

```
if ( function_exists('register_sidebar') ){
    register_sidebar(array(
        'name' => '側邊欄',
        'id' => 'sidebar',
        'description' => '顯示於每個網頁的右方。',
        'before_widget' => '<section id="%1$s"
class="sidebar-right">',
        'after_widget' => '</section>',
        'before_title' => '<h1 class="sidebar-title">',
        'after_title' => '</h1>'
    ));
}
```

函式及參數說明：

- register_sidebar() — 註冊選單之函式。
- 'name' — 模組名稱。
- 'id' — 模組 ID。
- 'description' — 模組說明。
- 'before_widget' — 在模組前的標籤。
- 'after_widget' — 模組後的標籤。
- 'before_title' — 模組名稱前的標籤。
- 'after_title' — 模組名稱後的標籤。

顯示於 sidebar.php 上

註冊完模組之後，你可以先在裡面新增幾個如最新文章、搜尋，這樣才會東西輸出：

接著，我們在 sidebar.php 加入以下代碼：

```
<aside id="sidebar">
    <?php dynamic_sidebar('sidebar'); ?>
</aside>
```

接著，在 style.css 加入一些樣式碼吧！

```
/* Sidebar */
.sidebar-right {
    margin: 30px 0;
    padding: 10px 15px;
    background: #FFF;
}
```

```
.sidebar-title {  
    font-size: 24px;  
    line-height: 1.8;  
    margin: 10px 0;  
    color: #666;  
}
```

現在，整個網頁長成這樣：



WordPress 佈景 single.php

現在，要開始來做一個網站也很重要的部分，那就是文章內頁囉！一個部落格中最重要的就是它的文章內容，如果文章排版的很乾淨、沒有雜七雜八的東西的話，那麼對文章來講是大大加分（當然文章的內容也很重要。）！

在文章內頁中，有一些代碼和 index.php 有些雷同，所以已經說明過的函式我們就不再講。而在 single.php 中，在文章的後面還有迴響區。另外，像顯示作者、插入 Google AdSense 廣告等實作，將會放在番外篇，讓這一節不會太過於冗長，想先看的讀者可以先偷偷的去看一下番外篇。

基本格局

基本格局，有部分和 index.php 相同。

```
<?php get_header(); ?>
<div class="content">
    <div class="article">
        <?php while ( have_posts() ) : the_post(); ?>
            <article class="article-content">
                <h1 class="article-title"><?php
the_title(); ?></h1>
                <div class="article-meta">
                    <span><?php the_time('n / j,
Y'); ?></span><span> / </span>
                    <span><?php the_category(' ,
'); ?></span><span> / </span>
                    <span><?php the_tags(' ', ' ', ' ',
' '); ?></span>
                </div>
                <?php the_content(); ?>
                <div class="clearfix"></div>
            </article>
            <?php endwhile; ?>
        </div>
        <div class="sidebar">
            <?php get_sidebar(); ?>
        </div>
    </div>
    <?php get_footer(); ?>
```

相似度相當高呢！呵呵，我們拿掉了什麼？頁碼。

迴響區塊嵌入

接著，我們要在文章之後插入顯示迴響的函式。在 `</article>` 後加入：

```
<div id="comments">
    <?php comments_template(); ?>
</div>
```

■ `comments_template()` — 載入 `comments.php` 的內容。

現在你打開來看是還沒有迴響的，因為我們還沒有製作 `comments.php`。

WordPress 佈景 page.php

接下來則是和 single.php 有九十五分神似的 page.php 了，page.php 是顯示頁面的檔案，而通常會放如聯絡資訊、關於我們、作品集等等資訊。

在 page.php 中，我們的基本格局和 single.php 幾乎一樣，去掉發表時間、分類與標籤，但不新增迴響樣板。

如果你想讓頁面與文章有不一樣的感覺，你可以自己修改。

基本格局

與 single.php 大致相同，但將發表時間、分類與標籤拿掉。（因為頁面沒有分類與標籤，也不需要顯示發表時間）

```
<?php get_header(); ?>
<div class="content">
    <div class="article">
        <?php while ( have_posts() ) : the_post(); ?>
            <article class="article-content">
                <h1 class="article-title"><?php
the_title(); ?></h1>
                <?php the_content(); ?>
                <div class="clearfix"></div>
            </article>
        <?php endwhile; ?>
    </div>
    <div class="sidebar">
        <?php get_sidebar(); ?>
    </div>
</div>
<?php get_footer(); ?>
```

完成。現在頁面長得像這樣：

樂在設計

Gallery

Works

About

Contact

範例頁面

這是一個範例頁面。它和網誌文章不同，因為它的頁面位置是固定的，同時會顯示於你的網誌導覽選單（大多數的佈景主題）。大多數的人會新增一個「關於」頁面向訪客介紹自己。它可能類似下面這樣：

嗨！你好！白天我是一位單車快遞員，晚上則是個有抱負的演員，這是我的網誌。我居住在台灣高雄，養了一隻名為 Jack 的狗。

...或像這樣：

XYZ Doohickey Comany 成立於 1971 年，公司成立以來，我們一直向市民提供高品質的 doohickies。我們位於台北市，有超過 2,000 名員工，對台北市政府有著相當大的貢獻。

作為一個新的 WordPress 使用者，你可以前往[你的控制台](#)刪除這個頁面，並建立一個屬於你的全新內容。祝使用愉快！

近期文章

- [Hello world! 哈囉！](#)
- [Hello world! 哈囉！](#)
- [Hello world! 哈囉！](#)

近期迴響

- admin 在 [Hello world! 哈囉！](#)
- admin 在 [Hello world! 哈囉！](#)
- admin 在 [Hello world! 哈囉！](#)
- admin 在 [Hello world! 哈囉！](#)
- WordPress 彙在 [Hello world! 哈囉！](#)

WordPress 佈景 comments.php

Comments.php 算是整個佈景裡頭有一點複雜的地方，但是做出來一個清爽的迴響介面可是會非常棒呢！

需要密碼判斷

首先，我們先寫一個判斷式，為需要密碼才能查看迴響的程式碼。

```
<?php
if ( post_password_required() )
    return;
?>
```

上段程式碼 post_password_required() 就是用來判斷是否需要密碼才能觀看迴響。

顯示迴響部分

接下來是顯示迴響的部分，這部分複雜的是在有頗多的判斷式。

```
<h3 class="comment-title"><?php comments_number('沒有迴響', '迴響 (1)', '迴響 (%)' );?></h3>
<?php if ( have_comments() ) : ?>

    <ol class="comment-list">
        <?php
wp_list_comments('type=comment&callback=displaycomments'); ?>
    </ol>
    <div class="clearfix"></div>
    <div class="pagenavi">
        <?php paginate_comments_links('prev_text=Prev Comments&next_text=Next Comments');?>
    </div>

<?php else : ?>
```

```

<?php if ( comments_open() ) : ?>
    <p>本文還沒有迴響，快來搶頭香！</p>
<?php else : ?>
    <p class="nocomments">本文不開放迴響。</p>
<?php endif; ?>

<?php endif; ?>

```

第一行的 `comments_number()` 是用來顯示迴響數目的函式，而參數分別是沒有迴響時顯示的文字，一個迴響時顯示的文字，兩個以上迴響時顯示的文字。

而第二行，有一個 `if (have_comments())` 的判斷式，而如果成立就會進入第三行到 `else` 之前的顯示迴響部分。這邊，有兩個要注意的地方：

- `wp_list_comments('type=comment&callback=displaycomments');` — 顯示出迴響的函式。在這邊有一個 `callback= displaycomments` 的參數，我們可以新建 `displaycomments` 函式來調整迴響的顯示格式。稍後將介紹 `displaycomments` 自訂函式。

另外 `type=comment` 的意思是只顯示迴響部分，而 `pingback` 與 `trackback` 則不顯示。

- `paginate_comments_links('prev_text=Prev Comments&next_text=Next Comments');` — 這部分則是當迴響超過控制台設定的單頁數目時，會顯示上一頁迴響及下一頁迴響。`prev_text` 及 `next_text` 則是設定顯示之文字。

接下來 `else` 之後的程式碼就是當目前該文章沒有迴響時會進來這裡，而沒有迴響有兩種情況，一種是沒有人留言，另外一種是不開放留言。所以我們可以使用 `if (comments_open())` 來判斷是否有開放迴響，有的話就是沒有人留言，沒有的話就是不開放迴響。

最後記得要補上 `endif;`。

新建 `displaycomments` 函式

先說明一下，`displaycomments` 函式並不是 WordPress 內建的函式，而是我們自己新增的函式。這個函式的用途是可以自訂每則迴響之顯示格式，我們先在 `functions.php` 加入以下的程式碼，再來解說。

```

function displaycomments($comment, $args, $depth){
    $GLOBALS['comment'] = $comment;
    ?>
    <li class="comment-list-box">

```



```

<div class="comment-author">
    <?php echo get_avatar( $comment, 40 ); ?>
</div>
<div class="comment-fn">
    <?php printf(__('<span class="fn">%s</span>'),
get_comment_author_link()) ?>
</div>
<div class="comment-meta">
    <?php printf(__('%1$s @ %2$s'),
get_comment_date(), get_comment_time()) ?> <?php
edit_comment_link() ?>
</div>
<?php if ($comment->comment_approved == '0') : ?>
    <em class="comment-approved">你的迴響正在審核中。
</em>
<?php endif;?>
<?php comment_text() ?>
<?php comment_reply_link(array_merge( $args,
array('depth' => $depth, 'max_depth' =>
$args['max_depth']))) ?>
<?php }

```

- `get_avatar($comment, 40);` — 這個函式是用來顯示出迴響留言者的大頭貼，而參數 `$comment` 表示是迴響用的，40 代表大小。
- `printf(__('%s'), get_comment_author_link())` — 這部分是顯示留言者名稱，如果有網站的話會一併加上連結。
- `<?php printf(__('%1$s @ %2$s'), get_comment_date(), get_comment_time()) ?> <?php edit_comment_link() ?>` — 這部分是顯示留言時間，`%1$s @ %2$s` 為格式。而 `edit_comment_link()` 則是當該留言者是會員時顯示編輯之連結。
- `if ($comment->comment_approved == '0')` 是用來判斷迴響是否有通過審核，若尚未審核則顯示正在審核中。
- `comment_text()` 就是迴響的內容啦！
- `comment_reply_link(array_merge($args, array('depth' => $depth, 'max_depth' => $args['max_depth'])))` — 這是用來顯示回覆的連結，而後面的參數是為了要讓階層功能正常，不可以省略。

留言表單部份

```
<?php comment_form(); ?>
```

由於 WordPress 有內建顯示迴響表單的函式，所以只要一行就能搞定了！

但是預設會顯示這兩句話：「你的電子郵件位址並不會被公開。 必要欄位標記為 *」與「你可以使用這些 HTML 標籤與屬性： ……」，如果不想要顯示的話，可改成這樣：

```
<?php  
comment_form("comment_notes_after=&comment_notes_befo  
re="); ?>
```

這樣就可以讓那兩句話消失。而 comment_notes_before 代表第一句話，comment_notes_after 代表第二句話。

迴響樣式碼

迴響樣式碼算是最難寫的部分，不過寫得好的話感覺會很棒。你可以特別去看一下寫法，有使用到 position 的功能。

```
/* Comments */  
#comments {  
    margin: 30px;  
    padding: 20px;  
    background: #fff;  
}  
#comments li {  
    list-style: none;  
}  
#comments li.comment-list-box {  
    padding: 10px 0 20px 55px;  
    margin: 0;  
    margin: 20px 0;  
    border-bottom: 1px solid #eeeeee;
```

```
    position: relative;
}
#comments li.comment-list-box .comment-author img {
    position: absolute;
    left: 0;
    float: left;
    margin: 0px 10px 5px 0px;
    border-radius: 50%;
    border: 1px solid #cccccc;
}

#comments li.comment-list-box a.comment-reply-link {
    border: 1px solid #dddddd;
    padding: 5px 13px;
    color: #888888;
    border-radius: 20px;
    font-size: 13px;
    display: inline-block;
}
#comments li.comment-list-box
a.comment-reply-link:hover {
    background: #eeeeee;
    color: #888888;
    text-decoration: none;
}
#comments ul.children {
    padding: 0;
    margin: 0;
}
#comments ul.children li {
    border-bottom: none;
    margin-bottom: 0;
    padding-bottom: 0;
}
#comments p {
    line-height: 1.48;
    margin: 10px 0;
}
#comments input[type="text"] {
    width: 30%;
}
```

```
#comments .comment-meta {
    font-size: 13px;
    line-height: 28px;
    color: #999999;
}
.cancel-comment-reply {
    line-height: 1.48;
}
.comment-list {
    margin: 10px 0 40px 0;
    padding: 0;
}
input[type="submit"] {
    padding: 7px 12px;
    border: 1px solid #888;
    background: #FFF;
    border-radius: 3px;
    font-size: 15px;
    cursor: pointer;
    outline: none;
}
input[type="submit"]:focus, input[type="submit"]:hover {
    background: #666;
    color: #FFF;
}
input[type="text"] {
    outline: none;
    padding: 7px 5px;
    border-radius: 3px;
    border: 1px solid #EEE;
    font-size: 15px;
}
input[type="text"]:focus {
    border-color: #A5CFE7;
    -webkit-transition: border-color 0.2s ease-in;
}
textarea {
    width: 95%;
    min-height: 160px;
    border: 1px solid #EEE;
```

```
border-radius: 3px;
padding: 8px;
outline: 0;
font-size: 15px;
}
textarea:focus {
border-color: #A5CFE7;
-webkit-transition: border-color 0.2s ease-in;
}
```

（洋洋灑灑兩頁半的 CSS 阿...）

另外，我們在這邊順便將 input 與 textarea 的樣式做好。這樣，整個迴響區域就完成囉！恭喜！

目前文章迴響的地方長得像這樣：

WordPress 佈景主題開發指南：歡迎來到 WordPress 主題的開發世界！用 WordPress 佈景主題開發 WordPress 佈景主題，然後開始你的部落客生活！

迴響 (1)



Test

2013 年 07 月 27 日 @ 13:30:15

這是一則測試迴響。

回覆

發表迴響

你的電子郵件位址並不會被公開。必要欄位標記為 *

名稱 *

電子郵件 *

個人網站

迴響

你可以使用這些 HTML 標籤與屬性： <abbr title=""> <acronym title=""> <blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <strike>

張貼迴響

WordPress 佈景 archive.php

Archive.php 是用來顯示彙整的文章，如列出標籤為某標籤的所有文章，分類為某分類的全部文章，也可以用依年分顯示文章。

有些佈景是將標籤、分類另外獨立出 tag.php 與 category.php 兩個檔案，而在這邊，我們只用 archive.php 來達成，比較方便一點。

基本格局

其實也是跟前面顯示文章的地方相當類似：

```
<?php get_header(); ?>
<div class="content">
    <div class="article">
        <?php while ( have_posts() ) : the_post(); ?>
            <article class="article-content">
                <h1 class="article-title"><a href="<?php
the_permalink(); ?>"><?php the_title(); ?></a></h1>
                <div class="article-meta">
                    <span><?php the_time('n / j,
Y'); ?></span><span> / </span>
                    <span><?php the_category(' ,
'); ?></span><span> / </span>
                    <span><?php the_tags(' ', ' ', ' ,
'); ?></span>
                </div>
                <?php the_content(); ?>
                <div class="clearfix"></div>
            </article>
        <?php endwhile; ?>
    </div>
    <?php wp_pagenavi(); ?>
    <div class="sidebar">
        <?php get_sidebar(); ?>
    </div>
</div>
<?php get_footer(); ?>
```

而要怎麼讓使用者知道這是一個彙整的頁面？在後面 breadcrumb 的地方會再

進行補充的動作。

WordPress 佈景 search.php

Search.php 會顯示 WordPress 內建的搜尋系統之搜尋解果，當使用者於站內搜尋框搜尋後，將會使用 search.php 來顯示結果。

當然，程式碼與上一章也是非常類似，不過多了一個 have_posts() 來判斷是否有搜尋到文章，沒有的則提示沒有結果，請使用別的關鍵字再搜尋一次。

```
<?php get_header(); ?>
<div class="content">
    <div class="article">
        <?php if ( have_posts() ) : ?>
            <?php while ( have_posts() ) : the_post(); ?>
                <article class="article-content">
                    <h1 class="article-title"><a
href="<?php the_permalink(); ?>"><?php
the_title(); ?></a></h1>
                    <div class="article-meta">
                        <span><?php the_time('n / j,
Y'); ?></span><span> / </span>
                        <span><?php the_category(' ,
'); ?></span><span> / </span>
                        <span><?php the_tags(' ', ' ', ' ',
' '); ?></span>
                    </div>
                    <?php the_content(); ?>
                    <div class="clearfix"></div>
                </article>
            <?php endwhile; ?>
        <?php else : ?>
            <article class="article-content">
                <h1>搜尋結果</h1>
                <p>很抱歉，找不到你所搜尋的文章，你可以試著用
其他關鍵字再次搜尋。</p>
                <?php get_search_form(); ?>
            </article>
        <?php endif; ?>
    </div>
    <?php wp_pagenavi(); ?>
</div class="sidebar">
```

```
<?php get_sidebar(); ?>
</div>
<?php get_footer(); ?>
```

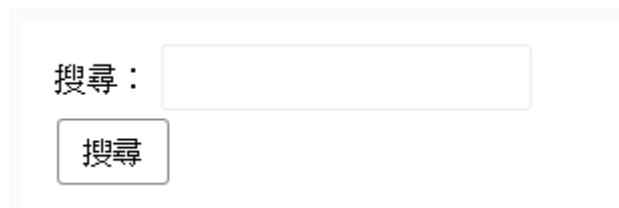
■ `get_search_form` — 顯示迴響表單（`searchform.php`）。
這樣，顯示搜尋結果頁也完成囉！

WordPress 佈景 searchform.php

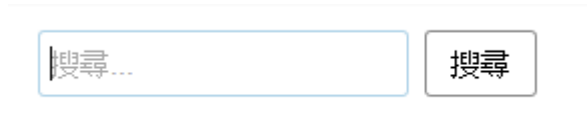
Searchform.php 顧名思義就是搜尋的表單，沒錯，我們現在就要來製作搜尋的表單囉！

由於 WordPress 內建的搜尋表單不是很好，在搜尋框之前還有一個「搜尋：」，不是很好看，所以我們可以使用 searchform.php 來更換搜尋表單的格式。

這是原本 WordPress 內建的搜尋表單：

A screenshot of the default WordPress search form. It consists of a text input field preceded by the label "搜尋：" (Search:). Below the input field is a submit button with the text "搜尋" (Search).

使用 searchform.php 修改之後：

A screenshot of the modified search form using the searchform.php template. The text "搜尋：" has been removed, and the input field now contains a placeholder text "搜尋..." (Search...). The submit button remains "搜尋" (Search).

是不是變得比較好看了呢？要達成這樣的效果，就在 searchform.php 加入下面的代碼就可以了：

```
<form role="search" method="get" id="searchform"
action="php echo home_url( '/' ); ?&gt;"&gt;
    &lt;input type="text" value="" name="s" id="s"
placeholder="搜尋..." /&gt;
    &lt;input type="submit" id="searchsubmit" value="搜尋"
/&gt;
&lt;/form&gt;</pre
```

其中，placeholder 就是在上圖的表單中提示搜尋… 的字樣那個。這樣，搜尋表單就完成囉！

WordPress 佈景 404.php

404.php 顧名思義就是當出現 404 Not Found 錯誤時，會顯示出的網頁。而 404.php 通常會提供回到首頁的連結、搜尋表單或者是隨選文章等等。在這章中，我們將所有分類列出，並提示使用者可回到首頁。

寫出來程式碼長這樣：

```
<?php get_header(); ?>
<div class="content">
    <div class="article">
        <article class="article-content">
            <h1>404 找不到網頁</h1>
            <p>
                抱歉，找不到您所要的頁面，或許已經移除、暫時關閉或發生錯誤。
            </p>
            <p>
                您可經由<a href="<?php bloginfo('url');?>" title="回到首頁">回到首頁</a>或以下分類及時間找到您所要的內容，或以下搜尋框：
            </p>
            <?php get_search_form(); ?>
            <h2>依分類查詢</h2>
            <ul class="errorlist">
                <?php
wp_list_categories('orderby=ID&show_count=1&use_desc_for_title=0&title_li=&style=list');?>
                </ul>
            </article>
        </div>
        <div class="sidebar">
            <?php get_sidebar(); ?>
        </div>
    </div>
<?php get_footer(); ?>
```

- `wp_list_categories('orderby=ID&show_count=1&use_desc_for_title=0&title_li=')`
 - 列出所有分類。
 - Orderby — 顯示順序之依據，show_count — 是否顯示

分類下文章數目，`use_desc_for_title` 一是否顯示分類描述，`title_li` 一分類標題。詳細參數可參考：[WP Codex— WP list category](#)。
這樣，`404.php` 就完成囉！

WordPress 佈景 functions.php

Functions.php 可以放入許多在佈景中要使用的自訂函式，而佈景內每一個檔案都可以使用。為什麼要把某項功能寫成函式？

在製作一個佈景中，可能某種功能會被重複利用好幾次，這時候將同一段程式碼重打是非常沒有效率的。而通常，我們會將它寫成一個函式，並且要將函式分門別類歸好，不過因為在佈景中的函式通常不多，我們可以直接通通塞在 functions.php 裡。

有些比較複雜的佈景，可能還會將不同功能的函式再分門別類歸屬到不同的檔案中，優點是將來要做修改或開發，可以直接針對某個檔案中的函式進行修改，不需要大費周章去很多檔案找來找去還不知道該如何下手。

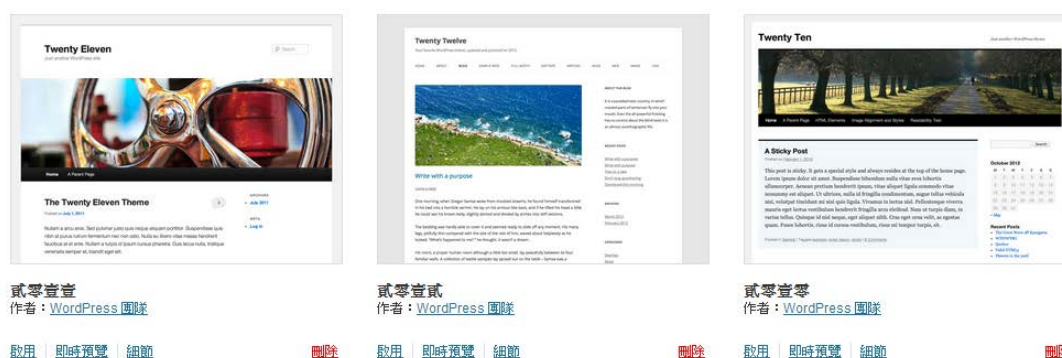
此外，WordPress 有內建非常多的函式，要好好利用！

WordPress 佈景 screenshot.png

在 WordPress 控制台 > 外觀中，會列出目前有的佈景，而通常在上面還可以附帶一張佈景縮圖。而這張圖片的檔名很簡單，是 screenshot.png。也就是說，你要在佈景列表那顯示出一張佈景縮圖，就在佈景目錄中放上你想顯示的圖片並將檔名設為 screenshot.png。

如 WordPress 內建之佈景縮圖：

可用佈景主題



附註一下，通常佈景縮圖會用兩種類型的圖片：一是佈景顯示效果截圖，另外一種是佈景 Logo。

番外篇

WordPress Breadcrumb 麵包屑製作

第一次看到 Breadcrumb 的人一定會有個疑問：這能吃嗎？

Breadcrumb 其實顯示出來就是類似「首頁 > 程式設計 > WordPress 介紹」這樣，可以讓使用者知道目前所在的網頁位置。算是個不起眼但不可缺少的小設計。

而我們這次要將 Breadcrumb 的功能寫成函式，方便每個檔案都能輕鬆使用。

```
function breadcrumb_init(){
    global $post;
?>
    <ul class="breadcrumb">
        <li itemscope
itemtype="http://data-vocabulary.org/Breadcrumb">
            <a href="<?php bloginfo('url');?>"
itemprop="url" title="<?php bloginfo('name');?>">
                <span itemprop="title"><?php
bloginfo('name');?></span></a>
            </li>
        <?php
        if( is_single() ) {
            foreach ( get_the_category() as $category) {
                echo '<li itemscope
itemtype="http://data-vocabulary.org/Breadcrumb">';
                echo '<span class="divider">></span> <a
href="" . get_category_link($category -> term_id) . "'
itemprop="url" title=" . $category -> cat_name . '> <span
itemprop="title">' . $category -> cat_name . '</span>
</a>';
                echo '</li>';
            }
        } else { ?>
            <li itemscope
itemtype="http://data-vocabulary.org/Breadcrumb"
class="active">
                <span class="divider">></span>
                <span itemprop="title">
                <?php
```

```

        if ( is_category() ) {
            echo single_cat_title();
        } elseif ( is_tag() ) {
            echo single_tag_title( '', true);
        } elseif ( is_day() ) {
            the_time( get_option('date_format') );
        } elseif ( is_month() ) {
            the_time( 'F, Y' );
        } elseif ( is_year() ) {
            the_time( 'Y' );
        } elseif ( is_page() ) {
            the_title();
        } ?></span>
    </li>
    <?php } ?>
</ul>
<?php
}

```

這部分使用到的函式比較多是 `is_` 開頭的，都是用來判斷目前使用者所在的頁面來顯示不同的文字。

再來，再加上一點 CSS：

```

/* Breadcrumb */
.breadcrumb {
    overflow: hidden;
    white-space: nowrap;
    margin: 0;
    background: transparent;
    font-size: 13px;
    color: #999999;
    padding: 0;
    margin-left: 50px;
}

.breadcrumb li {
    margin: 0;
    display: inline-block;
    text-shadow: 0 1px 0 white;
}

```

```
.breadcrumb i {
    vertical-align: middle;
    opacity: 0.5;
    margin: 0px 4px 0 0;
}

.breadcrumb a {
    color: #999999;
}

.breadcrumb .divider {
    padding: 0 5px;
    color: #cccccc;
}
```

之後，如果要顯示出 Breadcrumb 可以直接使用此函式：breadcrumb_init();。如，你可以在 archive.php、single.php、page.php 的 <div class="article"> 之後加入

```
<?php breadcrumb_init(); ?>
```

之後就可以顯示出麵包屑囉！



WordPress 作者介紹使用函式實作

在文章頁面中，通常在文章結束後會顯示文章的作者資訊，也有作者的自我介紹。通常還會附上 Facebook 的個人頁面網址。而這次我們將使用函式來完成這項功能。

在控制台個人資訊新增選項

首先，我們可以在控制台 > 個人資訊裡面新增選項，讓作者的 Facebook 網址等等可以直接透過控制台修改。

```
function add_user_profile( $contactmethods ) {
    $contactmethods['google'] = 'Google+ 個人網址';
    $contactmethods['facebook'] = 'Facebook 個人網址';
    $contactmethods['description_url'] = '個人介紹頁';
    return $contactmethods;
}
add_filter('user_contactmethods','add_user_profile',10,1);
```

WordPress 有一個非常好用的地方，我們可以自己建立一個 function 然後使用 add_filter() 函式，如本例，在個人資料（user_contactmethods）就會顯示出更多選項。

而這個新增的規則就是 \$contactmethods['ID'] = '選項名稱'。

寫成 function

當然，接下來就要寫成 function 了，一番折磨後像這樣：

```
function article_author(){
    global $post;
    ?>
    <section class="article-author">
        <div class="author-avatar"><?php echo
get_avatar( get_the_author_meta('ID'), 100);?></div>
        <div class="author-text">
            <h3 class="author-name"><?php
```

```

the_author());?></h3>
    <p class="author-description"><?php
the_author_meta('description');?></p>
    <div class="author-social">
        <?php if
( get_the_author_meta( 'google' ) ): ?>
            <a href="<?php
the_author_meta('google');?>?rel=author" title="我的
Google+">Google+</a>
        <?php endif; if
( get_the_author_meta( 'facebook' ) ): ?>
            | <a href="<?php
the_author_meta('facebook');?>" title="我的
Facebook">Facebook</a>
        <?php endif; if
( get_the_author_meta( 'description_url' ) ): ?>
            | <a href="<?php
the_author_meta('description_url');?>" title="<?php
the_author();?> 個人介紹">個人介紹</a><?php endif;?>
    </div>
</div>
</section>
<?php
}

```

在這邊，有幾個函式要說明一下：

- `get_avatar` — 取得大頭貼，而在迴響的地方我們是用 `$comment`，但在文章裡我們可以直接用作者的 ID 來存取作者的大頭貼，而 100 則是大小。非常方便。
- `get_the_author_meta(ID)` — 就是存取在控制台個人資訊的選項囉！
- `the_author_meta(ID)` — 顯示控制台個人資訊的選項。
- `the_author_meta('description')` — 顯示作者介紹。

Style

剩下的就是 CSS 的部分囉！希望你在學會之後，還能改出更好的樣式出來 ^__^

```

.article-author {
    width: 100%;
    float: left;
}

```

```
margin: 20px 0;
border-top: 1px solid #EEE;
padding: 20px 0;
border-bottom: 1px solid #EEE;
}
.author-avatar {
width: 100px;
float: left;
}
.author-avatar .avatar {
border-radius: 50%;
}
.author-text {
float: left;
width: 480px;
padding-left: 20px;
}
.author-name {
float: left;
width: 100%;
}
.author-description {
float: left;
width: 100%;
}
.author-social {
float: right;
width: 100%;
text-align: right;
}
```

接著，你就可以在 `single.php` 的 `</article>` 前面加上

```
<?php article_author(); ?>
```

就會在文章頁面顯示出作者頭像囉！

Hello world! 哈囉！

7 / 20, 2013 / Fun /

歡迎來到 WordPress。這是你的第一篇文章。編輯或者刪除本篇文章，然後開始你的部落客生活！歡迎來到 WordPress。這是你的



Yuxin

Hi！這是自我介紹顯示的地方。

[Google+](#) | [Facebook](#) | [個人介紹](#)

沒有迴響

WordPress 佈景文章縮圖功能實作

在許多部落格、網站上，其文章列表的每一個文章旁邊，通常都會有一張在那篇文章之中的圖片。而這個功能最簡單可以利用 WordPress 內建的「特色圖片」(在編輯文章右側)，直接去抓取圖片並且顯示出來。

另外，有些佈景主題會去自動抓取文章中第一張圖片作為特色圖片，其方法也會在本章中介紹。

有一個問題，如果抓取出來每一篇圖片大小不一，排版很醜，用 HTML width 控制圖片又會變得很醜，怎麼辦？我們有兩種方法，第一種是使用 Timthumb，第二種是使用 [WordPress 內建的 Crop 功能](#)。不過目前較多佈景是使用 Timthumb，和 WordPress 內建的 Crop 功能差異就是儲存的地方一個是佈景內，另一個是 /wp-content/uploads。

開啟特色圖片功能

這個特色圖片的功能預設是沒有開啟的，現在我們要把他打開。

```
if ( function_exists( 'add_theme_support' ) ) {  
    add_theme_support( 'post-thumbnails' );  
}
```

語法很簡單，就是 add_theme_support 增加佈景支援功能→post-thumbnails 中譯特色圖片。接著你就會看到他出現了：



取得特色圖片圖片網址

取得特色圖片 WordPress ([WP Codex—Post Thumbnails](#))

有內建函式，首先是判斷有沒有特色圖片：

```
if ( has_post_thumbnail() ) { /*有特色圖片執行這裡*/ }
```

取得特色圖片：(含)

```
$img = get_the_post_thumbnail();
```

顯示特色圖片：(含)

```
the_post_thumbnail();  
// 相當於 echo get_the_post_thumbnail()
```

取得特色圖片圖片網址：(只有圖片路徑)

```
wp_get_attachment_url( get_post_thumbnail_id() );
```

抓取文章第一張圖片

而抓取文章第一張圖片的方式，也非常簡單，只要寫成一個函式就可以達成。另外，這個函式會先判斷目前有沒有設定特色圖片，沒有設定才會去抓第一張圖片，有設定就會直接回傳特色圖片路徑。

```
function get_feature_image(){
    global $post, $posts;
    $first_img = '';
    if ( has_post_thumbnail() ) {
        $first_img =
wp_get_attachment_url( get_post_thumbnail_id() );
    } else {
        ob_start();
        ob_end_clean();
        $output = preg_match('/< *img[^>]*src *=
*["\']?([^"\']*)/i', $post->post_content, $matches);
        $first_img = $matches[1];
    }
    return $first_img;
}
```

而抓取第一張圖片的方式其實很簡單，我們只要用 PHP 的 preg_match 用正規表達式去比對就可以了。另外，因為我們只要第一張圖片，所以不使用 preg_match_all。

使用 Timthumb 裁切圖片

Timthumb 是一個非常好用的圖片裁切工具，他可以依照你所給的圖片尺寸去做裁切。而要使用 Timthumb 之前，你需要先去下載 [Timthumb.php](#)。下載回來後，不要對 Timthumb.php 做修改，你可以放到佈景目錄裡頭。

另外，要新增一個資料夾名為 cache，提供給 Timthumb 放置裁切後的圖片。接著，如果你要在佈景的任何地方使用，你只要使用下面這一行語法：

```

```

再來，你只要把「裁切後寬度」和「裁切後高度」這兩個地方改成你要的大小即可。(不必加 px)。另外，`zc=1` 代表要裁切，預設從圖片中間開始裁切。

WordPress Og:Meta 使用函式實作

最近，Facebook 等社群網站的分享越來越頻繁，而如果要 Facebook 轉貼連結時顯示的文章縮圖、描述正確的話，就必須使用<meta property="og:title">等標籤，這麼一來你就可以控制 Facebook 顯示的縮圖和描述，換句話說就是能讓 Facebook 抓的資料正確。

另外，下面這些代碼最初是由[就是教不落的阿湯](#)分享出來的，後來經過一些小修改。

其實原理很簡單，還記得我們在 header.php 中提到的 wp_head(); 嗎？我們就要利用這個地方，讓 function 在這裡插入一些 og 標籤。

只要將這個 function 加入到 functions.php 就可以了。

```
function insert_fb_in_head() {
    global $post;
    if ( is_home() ) {
        echo '<meta property="fb:admins" content="管理員的 Facebook 帳號 ID" />';
        echo "\n";
        echo '<meta property="fb:app_id" content="網站 Facebook APP 的 ID" />';
        echo "\n";
        echo '<meta property="og:type" content="website"/>';
        echo "\n";
        echo '<meta property="og:title" content="' . get_bloginfo('name') . '" />';
        echo "\n";
        echo '<meta property="og:description" content="' . get_bloginfo('description') . '" />';
        echo "\n";
        echo '<meta property="og:url" content="' . get_bloginfo('url') . '" />';
        echo "\n";
        echo '<meta property="og:site_name" content="' . get_bloginfo('name') . '" />';
        echo "\n";
        echo '<meta property="og:locale" content="zh_tw">';
        echo "\n";
    }
}
```

```

    }
    if ( !is_singular() ) return;
    $post_excerpt = ( $post->post_excerpt ) ?
$post->post_excerpt : trim( str_replace( "\r\n", ' ',
strip_tags( $post->post_content ) ) );
    $description = mb_substr( $post_excerpt, 0, 160,
'UTF-8' );
    $description .= ( mb_strlen( $post_excerpt, 'UTF-8' )
> 160 ) ? '...' : '';
    echo "\n";
    echo '<meta property="fb:admins" content="管理員
的 Facebook 帳號 ID" />';
    echo "\n";
    echo '<meta property="fb:app_id" content="網站
Facebook APP 的 ID" />';
    echo "\n";
    echo '<meta property="og:title" content="' .
get_the_title() . '" />';
    echo "\n";
    echo '<meta property="og:description"
content="' . $description . '" />';
    echo "\n";
    echo '<meta property="og:type"
content="article"/>';
    echo "\n";
    echo '<meta property="og:url" content="' .
get_permalink() . '" />';
    echo "\n";
    echo '<meta property="og:site_name" content="' .
get_bloginfo('name') . '" />';
    echo "\n";
    echo '<meta property="og:image"
content="' . $img . '" />';
    echo "\n";
    echo '<link rel="image_src" type="image/jpeg"
href="' . get_feature_image() . '" />';
    echo "\n";
    echo '<meta property="og:locale"
content="zh_tw">';
    echo "\n";
}

```

```
add_action( 'wp_head', 'insert_fb_in_head', 10 );
```

新增完以後，你要先去 [Facebook Developer APP](#) 新增一個屬於你的網站的 APP，然後再把 APP ID 替換「網站 Facebook APP 的 ID」，你的 Facebook 帳號 ID 替換「管理員的 Facebook 帳號 ID」。(共四處)

■ `add_action('wp_head', 'insert_fb_in_head', 10);`

`add_action()` 這個函式可以說是 WordPress 廣為流行的原因之一，它可以讓開發者在佈景內、控制台內新增非常多的自定義選項、文字。第一個參數 `wp_head` 就是插入到哪個 WordPress 哪個函式中，`insert_fb_in_head` 就是我們自己新增的函式，會被 `wp_head` 拿去用。後面的 10 是重要程度，數字越小越重要。

WordPress 在文章 more 標記處插入廣告代碼

很多網站會在文章 More 標記處插入一個或兩個的 Google AdSense 廣告，而實現方法很簡單，我們只要利用拆字串的方法，把 More 標記的 HTML 找出來，在那邊插入廣告代碼就可以了！

```
function adsense_adder_at_more_tag($text) {  
if( is_single() ) :  
$ads_text = '  
  
請將廣告代碼貼在這裡  
  
';  
$pos1 = strpos($text, '<span id="more-');  
$pos2 = strpos($text, '</span>', $pos1);  
$text1 = substr($text, 0, $pos2);  
$text2 = substr($text, $pos2);  
$text = $text1 . $ads_text . $text2;  
endif;  
return $text;  
}  
add_filter('the_content',  
'adsense_adder_at_more_tag');
```

新增完，記得將「請將廣告代碼貼在這裡」替換成你的廣告代碼。這樣就完成囉！

後記

這一份 WordPress 佈景主題開發指南其實是定位在已經有程式基礎的人，想要快速進入 WordPress 佈景主題開發領域的人閱讀的。不管你閱讀完這一份指南之後覺得好或者不好，我都很謝謝你下載這一份指南回去閱讀，希望你能從這份指南中得到一些你原本不知道的東西，獲得成長，進擊！

關於作者

Yuxin

程式部落格：[樂在設計](#)

電子信箱：yuxin995@gmail.com

引註連結

1. [《WordPress 筆記》讓 Facebook 及 Google+ 正確抓取內容及指定圖片](#)
2. [WordPress Codex](#)
3. [讓 WordPress 自動抓內文前 200 字當 description，在 Google+ 轉貼文章時可以抓到正確前言](#)
4. [WordPress 筆記本](#)
5. [TimThumb 超強 PHP 縮圖程式，搞定 WordPress 縮圖功能](#)

※ 在本份指南中有使用到你寫的程式碼卻沒有在這邊嗎？拜託，一定要用上一頁的電子信箱聯絡我，我一定會將你的網站文章網址放上去。

WordPress 推薦/相關網站

1. [WordPress Codex](#)
2. [WordPress Tuts+](#)
3. [香腸炒魷魚](#)
4. [就是教不落](#)
5. [WordPress 筆記本](#)
6. [樂在設計](#)
7. [免費 WordPress 佈景主題教學電子書（免費資源網路社群）](#)

※你也有寫 WordPress 的教學文章，而且寫得不錯的嗎？歡迎透過電子信箱連絡我毛遂自薦，我會將你的網站加到上述列表中喔！

Changelog

- 2013.7.28 — 修正忘了寫呼叫 `wp_pagenavi()` ，以及 Breadcrumb 的 CSS。