

# 序

對想要知道 Ajax 正如何被應用到現實世界的開發者、設計者、和經理人，《Ajax 設計模式》是為你量身打造的一份參考資料。Ajax 是一個新名詞，描述行為豐富、類似桌面應用程式的 Web 應用程式，這些應用程式執行在標準 Web 瀏覽器上，不需外掛任何特別的插件。它的風行與普及，歸功於一些眾所矚目的 Ajax 應用程式，像是 Gmail，Google Maps，和 37signals 的 Basecamp。在企業界，Ajax 也越來越受歡迎。

本書的設計模式分成 4 類。基礎技術（Foundational Technology）設計模式概述開發 Ajax 所需要的原始技術，例如 XMLHttpRequest 物件。編程（Programming）設計模式呈現開發者一直在發掘的開發技術，這些技術確保他們的 Ajax 應用程式維護容易且效能良好。功能性和使用性（Functionality and Usability）設計模式是關於 Ajax 應用程式會遇到的使用者介面類型，以及所能達成的新功能。最後，開發（Development）設計模式闡述用以監控、偵錯、及測試 Ajax 應用程式的流程。

## 誰該閱讀這本書？

如果想要達成下列事項，你應該閱讀這本書：

- 瞭解 Ajax 能辦到什麼，不能辦到什麼，以及正如何被運用到現實世界裡。
- 加速理解 Ajax 的核心技術，例如 XMLHttpRequest，DOM，以及 JSON 等。
- 找出開發者正用以產生高品質 Ajax 架構，加速效能，及改善使用性的設計模式。

因為設計模式的參考性本質，本書可為各種背景的人所理解。想要在 Ajax 領域上快速提升的開發者，可從教學課程的章節（第二章）及基礎技術設計模式開始。已經有些 Ajax 經驗的人，可能從編程和開發設計模式獲益最多。擔任較非技術領域角色的人，可特別著重在功能性和使用性設計模式及真實世界的例子，看看 Ajax 能做到什麼程度，以獲取高階的觀點。

為能領會書中的技術性討論與程式碼範例，程式人員應該對網頁開發基礎有些經驗 - HTML，表單提交，伺服器端程式，等等。最好，你也應該知道一些 JavaScript，因為這本書不是要教你程式語言。但本書的撰寫確實有考慮到：許多讀者對 JavaScript 只有基本的熟悉度。

Ajax 主要是關於發生在瀏覽器裡的事，因此本書不假設你知道任何特定的伺服器端環境。在那些牽涉到伺服器端程式碼的地方，範例基本上以 PHP 為基礎，並且總是用語言中立（language-neutral）的術語作解釋。

## 誰不該閱讀這本書？

如果你沒有從事過任何網頁開發的工作，在一頭栽進 Ajax 和這些設計模式之前，或許應該找些基礎概念的介紹來看看比較好。

Ajax 開發牽涉到廣大範圍的技術領域，包括 HTML，XML，CSS，JavaScript 和伺服器端開發。這本書將幫助你理解 Ajax 如何與這些技術的每一項有關聯，通常怎樣結合在一起，但是，如果你指望徹底學習這些技術的任一項，或者你正在尋求有關瀏覽器細節的參考，建議你查閱更專門的參考資料。

## 關於範例

本書的所有範例（教學課程程式碼與 Ajax 設計模式程式碼）可從 <http://ajaxify.com/run> 下載。它們已被開發成與新近版本的 Firefox 和 IE 相容，並且在 Firefox 1.5 和 IE 6.0【註】上測試過。多數伺服器端程式碼需要 PHP 5 或更新版本，並在 PHP 5.0.4 上測試過。PHP 程式碼應該跑在任何能執行 PHP script 的標準 Web 伺服器上；Apache 1.3.33 被用於本書範例的測試。Wiki Demo 需要 MySQL，並在 4.1.14-max 的版本上測試過。關於範例安裝的資訊，請參閱附錄 B。

## 瀏覽器細節

實際上對 Ajax 的目的來說，所有 Ajax 設計模式都能在任何現今的標準瀏覽器上實作，討論的焦點應側重於一般性架構與使用性的議題，而不是瀏覽器的細節。但是，可攜性（portability）議題確實出現在某些案例上，並且在對實作設計模式具有關鍵性影響的地方被說明（如同本書第二部分，基礎技術設計模式，中的某些主題所示）。為不模糊焦點，多數討論仍局限於 IE 和 Firefox；描繪每個瀏覽器和版本的風格與手法，超出設計模式所要討論的範圍，為此，建議你參考更專門的資料。

## 本書組織

### 第一部分，介紹

最初幾章是設計模式的序曲。第一章〈Ajax 簡介〉概述 Ajax 現象並探討當前趨勢。第二章〈以設計模式為主軸的教學課程〉是快速入門 Ajax 和 Ajax 設計模式的教學課程。Ajax 設計模式背後的设计概念在第三章〈Ajax 設計：原則與設計模式〉討論，並伴隨設計模式本身的介紹。

### 第二部分，基礎技術設計模式

基礎技術設計模式描繪出任何 Ajax 應用程式核心所使用的「建構區塊」（building blocks）。Ajax 本身是一種設計模式，因此作為第一個設計模式，第四章〈Ajax 應用程式〉扮演所有 Ajax 設計模式的基礎。接下來幾章探討

●.....  
註 確保對所有主要瀏覽器（例如 Safari、Opera）具有完整兼容性會是比較好的，特別是對上線的系統而言。但在此，個別的 demo 並未特別考慮可攜性，每個 demo 主要是針對特定 Ajax 概念做介紹。

建立 Ajax 應用程式所涉及的 3 項核心技術。第五章〈顯示操控〉的相關模式是關於重繪使用者介面，第六章〈Web Remoting〉的相關模式收錄幾種與伺服器溝通的作法。第七章〈動態行為〉的相關模式則討論事件（event）—由使用者與時間驅動的事件。最後，第八章〈延伸技術〉解釋怎樣使用非標準技術，延伸 Ajax 應用程式。

### 第三部分，編程設計模式

編程設計模式聚焦於軟體的技術性品質，特別是可維護性，穩定性，及效能。這部分的第一個章節，即第九章，為 Web Service 設計提供幾個策略。此外，與 Web Remoting 有關，第十章〈瀏覽器與伺服器對話〉的相關模式探討瀏覽器與伺服器之間的資訊流動；例如，同步技術。第十一章〈DOM Population〉的相關模式涵蓋根據伺服器回應，為 DOM 填值的幾個處理策略。第十二章〈程式碼生成與重利用〉包含幾個關於可維護性和可攜性（portability）的一般性編程設計模式。最後，第十三章〈效能最佳化〉的相關模式不僅關於提升更新與溝通速度，並且為無可避免的延遲，改善使用者的操作經驗。

### 第四部分，功能性與使用性設計模式

功能性和使用性設計模式著重於 Ajax 應用程式的使用性。第十四章介紹各種構成 Ajax 介面的小組件（widget）。第十五章〈頁面架構〉的相關模式採取高階觀點，把焦點放在頁面佈置，內容分解，以及顯露伺服器端內容的技術。由於 Ajax 的普遍與廣受歡迎，網際網路上正興起一股視覺效果（visual effects）的熱潮，這些視覺效果的躍進甚至有點凌駕於傳統的桌面應用系統之上，第十六章〈視覺效果〉的相關模式描述最常見的視覺效果。這個部分的最後一章，第十七章〈功能性〉的相關模式是關於 Ajax 能達成的幾種新功能。

### 第五部分，開發設計模式

開發設計模式不是以先前設計模式相同觀點所談的「事物」，而是你能藉以協助開發工作的「流程」。第十八章〈診斷〉的相關模式幫助你偵錯及監測 Ajax 應用程式的健康狀況。第十九章〈測試〉的相關模式有助於進行各種層次的測試，並且隨著應用程式逐步演進，採用基於敏捷法（agile practice）的測試實務。

### 第六部分，附錄

共有 4 個附錄。附錄 A，最大的附錄，是一個 Ajax 相關框架與程式庫的清單，當中很多對實作本書設計模式，提供很大的幫助。附錄 B 是一組安裝程式碼範例的註釋。附錄 C 著墨於設計模式一般性理論的探討。最後，附錄 D 則列出本書的參考文獻。

# 本書體裁

## 斜體

用以表示設計模式名稱。

## 定寬體

用以表示程式範例或片段。

## 定寬粗體

用以表示重要的程式碼。

「瀏覽器」這個詞，是瀏覽器端上整個系統的簡化表示 — 不只是瀏覽器應用程式（例如 Firefox），還包括跑在上面的 Web 應用程式（例如 Google Maps）。當我特別指涉瀏覽器應用程式時，我通常會把它稱為「Web 瀏覽器」— 無論如何，從前後文來看，這是相當明顯的。同樣地，「伺服器」或「伺服器端」指的是伺服器上的所有東西 — 作業系統（例如 Linux），Web 伺服器（例如 Apache），以及執行在上面的特定應用程式的 script。

「ECMAScript」和「JScript」這兩個詞並未出現在本書，「JavaScript」被用來涵蓋這些名稱。

「類別」這個詞，被用來描述以類似物件導向系統類別的方式使用的 JavaScript 函式。

出現在每個設計模式裡的第一個圖形，是該設計模式的圖解概述。

本書所使用的共同首字母縮略詞：

- CSS：串接樣式表（Cascading Style Sheets）
- DHTML：動態 HTML（Dynamic HTML）
- DOM：文件物件模型（Document Object Model）
- HTML：超文字標記語言（Hypertext Markup Language）
- IE：Microsoft Internet Explorer
- XML：可延伸標記語言（eXtensible Markup Language）

## 程式碼範例的約定

設計圖解基本上根據統一塑模語言（UML）表示法繪成，但多數情況下，稍微不那麼正式。

在程式碼範例中，`.phtml` 後綴字用於輸出 HTML 的 PHP 檔；所有其他 PHP 檔（例如商業邏輯模組）則以 `.php` 結尾。

如第二章所述，`$()` 是 `document.getElementById` 的別名，用以減少程式碼的凌亂程度（受 *prototype* 程式庫啟發 — 見 <http://prototype.conio.net/>）。

為求清晰，某些程式碼範例被重新格式化過。

## 連絡我們

我們已盡力核驗本書所提供的資訊，儘管如此，仍不能保證本書完全沒有瑕疵，而網路世界的變化之快，也使得本書永不過時的保證成為不可能。如果讀者發現本書內容上的錯誤，不管是贅字、錯字、語意不清、甚至是技術錯誤，我們都竭誠虛心接受讀者指教。如果您有任何問題，以下是我們的聯絡資訊：

美商歐萊禮股份有限公司台灣分公司

電話：(02) 2709-9669      傳真：(02) 2703-8802

網頁：<http://www.oreilly.com.tw>

電子郵件：[mail@oreilly.com.tw](mailto:mail@oreilly.com.tw)

與本書有關的線上資訊（包括勘誤、範例程式、相關連結）：

原文書

<http://www.oreilly.com/catalog/ajaxdp/>

中文書

[http://www.oreilly.com.tw/product2\\_tip.php?id=a206](http://www.oreilly.com.tw/product2_tip.php?id=a206)

## 誌謝

在線上撰寫這本書正意謂著：在整個過程中，我接收到很多很棒的回饋，以及有許多該感謝的人。但另一種說法是，我非常可能遺漏掉某些人！如果情況真是如此，請來信告知，我會改正。

我的編輯，Simon St.Laurent，總是迅速對我的疑問作回應，並以有點不凡的方式將一本撰寫中的書籍處理得很棒。也謝謝本書的校稿人 Mary Brady 巨細靡遺的工作，以及插畫家 Rob Romano，感謝他的協助，讓本書內容一目了然。也要感謝歐萊禮接受 *Ajax Patterns* 這個網站，特別是讓我在線上撰寫這本書，還有內容毫無限制的部落格與網播。

這本書從 *Ajax Patterns* (<http://softwareas.com/ajax-patterns>) 上的部落格中成長，Thomas Baekdal 將 Ajax 使用性原則涵蓋進來的洞察力，啟發了最初發表的幾個設計模式 (<http://www.baekdal.com/articles/Usability/XMLHttpRequest-guidelines>)。也要謝謝 Brent Ashley，感謝他在最初的 Ajax 會議以及他的部落格裡關於設計模式的討論，很明顯地，這觸發了這個專案的線上工作最初的關注。這本書的想法也要大大歸功於 Jesse-James Garrett 與他的 Ajax 文章，沒有這些文章，就沒有這本書。

本書所有的審閱者，在整個寫作過程中，對線上版本提供了很多有價值的回饋，對本書的草稿版本，更提供許多巨細靡遺的意見。這些審閱者是：Cameron Shorter 和 Kevin Vinsen (ADI Limited, Mapbuilder)，還有他們的雇主，ADI，給予他們兩人主導審閱的時間；Jep Castelein (Backbase)，Daniel Czarnecki (Zoltak Consulting)，Tony Hill (Thomson Corporation)，以及 Alex Kirk (Blummy.com, WizLite.com)。另外，深深感謝 Urbana-Champaign 的伊利諾大學的 Software Architecture Group (SAG) 的成員，感謝他們主導一些關於設計模式、內容豐富的討論，並將討論內容的聲音檔公開在網路上 (<http://www.softwareas.com/sag-ajax-patterns-review-1>) — 這些寶貴的意見最後都融入本書的內容裡。領導這個團體的 Ralph Johnson，提名 *Ajax Patterns* 作為審閱的對象，還有 Brian Foote 在整個過程中讓我獲得最新資訊。我也感激加入 AjaxPatterns.org Wiki 公眾部分的每個人，他們透過電子郵件，部落格，及 Wiki 上的討論，提供線上草稿的回饋。

因為設計模式是從很多現有應用程式中發掘出來的，我想感謝所有這些範例的創造者，他們是真正的先鋒。由衷感謝對我解釋他們的設計面向的那些人：Kevin Arthur (Stream)，Richard Cowin 與 Bill Scott (OpenRico)，Donovan Preston (LivePage)，Jeremy Ruston (TiddlyWiki)，以及 Tino "Crisp" Zijdel (DHTML Lemmings)。設計模式也是奠基於許多文章，程式碼研究，想法，概念證明，以及直接的建議。許多設計模式之所以會出現在這裡，特別是因為下列作者和開發者（也會在個別的設計模式描述中提及）的貢獻：Julien Couvreur，James Dam，Abe Fettig (Twisted，Jotspot)，Chris Justus，Christopher Kruslicky，Alex Russell (Dojo，Jotspot)，Richard Schwartz，Mike Stenhouse，Joel Webber，以及 Chris Were。另外，有幾個以新聞為焦點的服務，在這些活動發生時，讓尋找到相關內容變得容易多（也幫助其他人找到 AjaxPatterns.org！）。這些資源有：Niner Niner 的 AjaxBlog.com，Chris Cornutt 的 AjaxDeveloper.org (Chris 目前在 Ajaxian.com)，Ajaxian.com 的 Dion Almaer，Ben Galbraith 與 Rob Sanheim，Shane Witbeck 的 AjaxMatters.com，Mike Papageorge 的

FiftyFourEleven.com，Brian Benzinger 的 SolutionWatch.com，以及 Mike Arrington 的 Techcrunch.com。

最後，但不是最不重要的，滿懷的感激要獻給我的家人，謝謝你們在整個寫作過程中的支持。