
前言

PHP 和 MySQL 的組合是設計動態的、使用資料庫的 web 時，最方便的工具，它們在面對其他一些更難學習的整合框架時，仍能保有自己的優勢。拜開放原始碼之賜，它是免費的，因此是極受歡迎的 Web 開發選項。

想要在 Unix/Linux，甚至在 Windows/Apache 平台上進行開發的人都必須掌握這些技術。而且，透過相關的技術，包括 JavaScript、React、CSS 與 HTML5，你可以建構符合 Facebook、Twitter 及 Gmail 等業界品質的網站。

對象

本書獻給想要做出讓自己滿意的動態網站的人，包括已經會製做靜態網站，或是會用 WordPress 等 CMS，但希望提升技術的網站管理員或圖形設計者，以及高中和大學學生、應屆畢業生，及自學者。

事實上，任何準備學習響應式 web 設計的人，都可以學到 PHP、MySQL、JavaScript、CSS 與 HTML5 的核心技術，也會學到 React 的基本知識。

使用本書的前提

本書認為你已經初步了解 HTML，至少可以寫出簡單的靜態網站，但不假設你學過 PHP、MySQL、JavaScript、CSS 或 HTML5，不過，如果你已經了解它們的話，你會學得更快。

本書架構

本書的章節是以特定的順序來編寫的，我們會先介紹所有的核心技術，然後帶領你在 web 開發伺服器上逐步安裝它們，讓你可以操作本書的範例。

在第一部分中，你會學到 PHP 程式語言的基礎知識，了解語法、陣列、函式及物件導向程式設計的基本概念。

讓你學會 PHP 之後，接下來將介紹 MySQL 資料庫系統，你會學到 MySQL 資料庫的結構，及如何建構複雜的查詢指令。

之後，你會學到如何結合 PHP 與 MySQL，並且整合表單及其他 HTML 功能，來建構你自己的動態網頁。接著，你將學習各種實用的函式、管理 cookie 與 session，以及維護高等級的安全性，來深入了解 PHP 與 MySQL 開發的細節。

在接下來幾章，你會建立紮實的 JavaScript 基礎，內容包括簡單的函式、處理事件、讀取文件物件模型、在瀏覽器進行驗證，以及處理錯誤。我們也會詳細地介紹如何使用 JavaScript 熱門的 React 程式庫。

了解三種核心技術之後，你會學到如何發出幕後 Ajax 呼叫，將網站變成高度動態的環境。

接下來，我們用兩章來介紹如何使用 CSS 來裝飾與排版網頁，接下來說明如何用 React 程式庫來大幅簡化開發工作。最後，我們介紹 HTML5 內建的互動式功能，包括地理定位、音訊、視訊，與 canvas。在此之後，你會用學過的所有技術來撰寫一個完整的程式，包含社交網站的完整功能。

在過程中，你會看到許多程式設計的優良做法和小提示，它們可以幫助你找到並解決難以檢測的程式錯誤。本書也有許多網站連結，可讓你進一步了解相關的主題。

本書編排方式

本書使用下列的編排方式：

純文字 (Plain text)

代表目錄標題、選項，與按鈕。

動態 web 內容簡介

全球資訊網（World Wide Web）是持續進化的網路，它在 1990 年代被創造出來時，目的只是為了解決特定的問題，但現在的概念與當時有很大的不同。當時的 CERN（歐洲量子物理實驗室，現在因為運行大型的強子對撞器而聞名）有一項尖端實驗產生了海量的資料，但他們無法將如此大量的資料傳送給世界各地的科學家。

當時 Internet 已經問世了，有數十萬台電腦與它連結，因此 Tim Berners-Lee（CERN 成員）發明一種方法，可讓科學家使用超連結的框架在那些電腦之間巡覽，這種方法後來稱為超文字傳輸通訊協定（Hypertext Transfer Protocol），即 HTTP。他也創造了一種標記語言，稱為超文字標記語言（Hypertext Markup Language），即 HTML。為了將它們結合在一起，他寫出史上第一個 web 瀏覽器與 web 伺服器

現在的我們可能認為那些工具都是理所當然的，但是在當時，家用數據機的使用者頂多只能撥接至電子布告欄，在上面和使用同一種服務的其他使用者進行通訊和交換資料。因此，你必須加入許多電子布告欄系統，才可以透過電子器材來與同事及朋友通訊。

但是 Berners-Lee 一舉改變所有狀況，到了 1990 年代中期，已經有三種主要的圖形 web 瀏覽器在角逐 500 萬名使用者的青睞。不過大家很快就發現到，網路顯然還缺少一些元素。在網頁上顯示好幾頁的文字以及連至其他網頁的超連結圖案的確是很棒的概念，但是它們並未充分發揮電腦與 Internet 的潛力，無法讓使用者瀏覽動態改變的內容。即使當時的文字已經可以捲動，也有動態 GIF 了，但 web 用起來依然是非常枯燥的體驗！

購物車、搜尋引擎，及社交網路已經改變我們使用 web 的方式了。本章會簡單地介紹 web 的元素，以及讓 web 具備豐富及動態體驗的軟體。



我們必須開始使用一些縮寫詞。我會先清楚地解釋它們再繼續往下談，但你先不用糾結於它們的意思，或名稱的意義，只要你繼續看下去就會豁然開朗。

HTTP 與 HTML：Berners-Lee 的基礎

HTTP 是一種通訊標準，負責管理在瀏覽器與 web 伺服器之間來回傳送的請求與回應。伺服器的工作是接收用戶端的請求，並試圖以有意義的方式來回應它，通常是傳遞對方請求的網頁，這也是它稱為伺服器（*server*）的原因。相對於伺服器的一方，自然就是用戶端（*client*）了，這個名詞可以用來代表 web 瀏覽器，以及運行它的電腦。

在用戶端與伺服器之間可能還有其他的設備，例如路由器、代理伺服器、通訊閘……等。它們分別發揮不同的作用，確保請求與回應在用戶端與伺服器之間被正確地傳遞。它們通常使用 Internet 來傳送這些資訊。有些中間設備也可以協助提升 Internet 的速度，它們的做法是將網頁或資訊存在本地的快取（*cache*）裡面，之後直接從快取將這些內容送給用戶端，而不是從來源伺服器一路傳遞它。

web 伺服器通常可以同時處理許多連結，當伺服器沒有和用戶端通訊時，也會花時間監聽進來的連結。如果有連結抵達，伺服器會回傳回應，以確認它已經收到了。

「請求 / 回應」程序

「請求 / 回應」程序基本上就是：網頁瀏覽器或其他的用戶端要求 web 伺服器傳送一個網頁給它，然後伺服器回傳該網頁。接下來，瀏覽器會負責顯示網頁（見圖 1-1）。

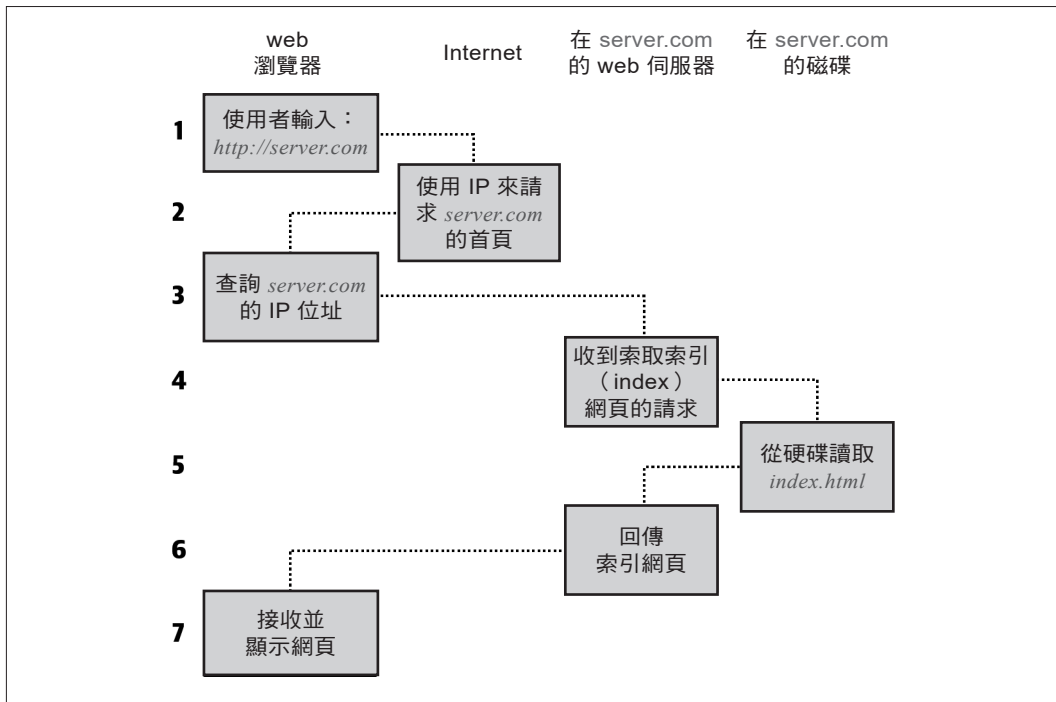


圖 1-1 「用戶端 / 伺服器」的基本「請求 / 回應」程序

以下是請求與回應程序中的每一個步驟：

1. 你在瀏覽器的網址列輸入 `http://server.com`。
2. 瀏覽器查詢 `server.com` 的網際網路通訊協定 (IP) 位址。
3. 瀏覽器發出一個索取 `server.com` 首頁的請求。
4. 這個請求經過 Internet 到達 `server.com` 網頁伺服器。
5. 收到請求的網頁伺服器在它的磁碟上尋找網頁。
6. 網頁伺服器取出該網頁，並將它回傳給瀏覽器。
7. 瀏覽器顯示網頁。

在一般的網頁裡面的每一個物件都會執行一次這種程序，無論該物件是圖片、內嵌的影片或 Flash 檔，甚至 CSS 模板都是如此。

注意在第二步中，瀏覽器會查詢 *server.com* 的 IP 位址。每一台連接 Internet 的機器都有一個 IP 位址，包括你的電腦。但是我們通常使用名稱來造訪網頁伺服器，例如 *google.com*。瀏覽器會額外諮詢一種 Internet 設備（網域名稱服務，DNS）來尋找伺服器的 IP 位址，再用 IP 位址來與電腦通訊。

動態網頁的程序比較複雜，因為這種網頁可能也會使用 PHP 與 MySQL（見圖 1-2）。例如，你可能會按下一張雨衣圖片。接著，PHP 會用標準資料庫語言 SQL（本書會教它的許多指令）來建立一個請求，並將這個請求送給 MySQL 伺服器。MySQL 伺服器會回傳那件雨衣的資訊，接著 PHP 碼會將它包在某些 HTML 裡面，再由伺服器送給你的瀏覽器（見圖 1-2）。

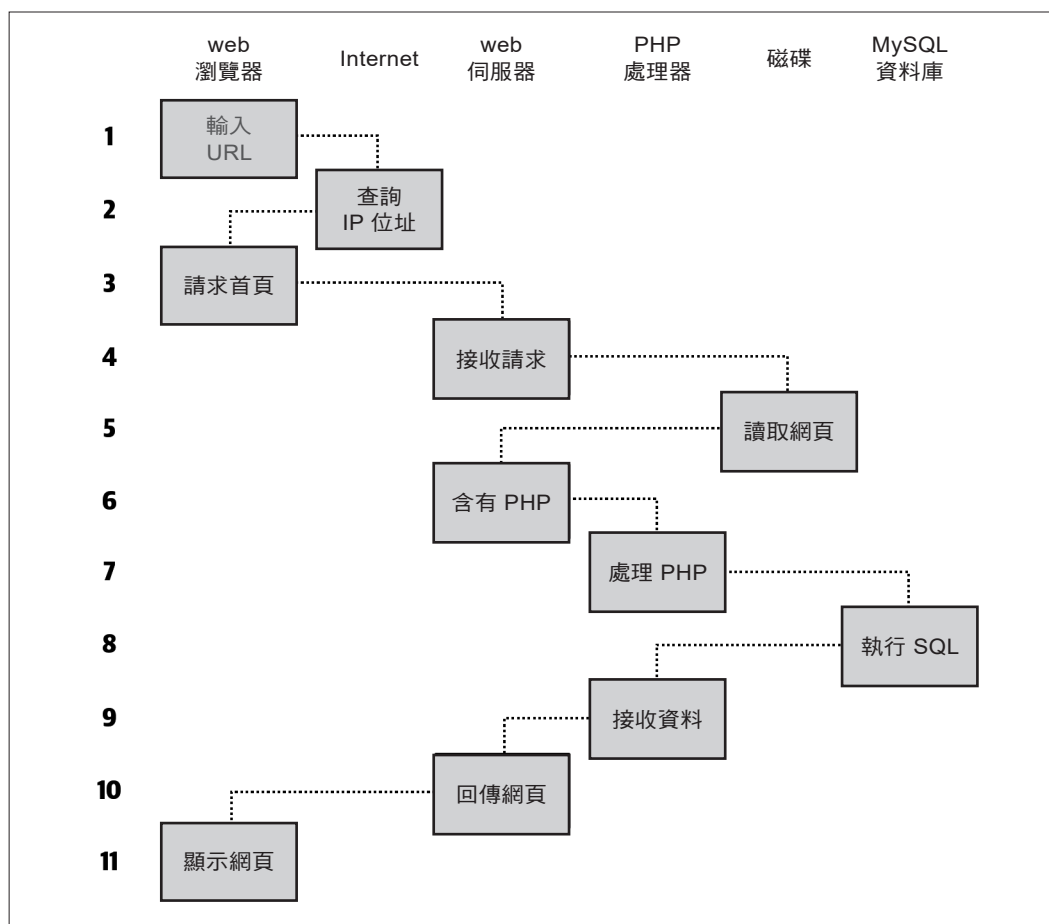


圖 1-2 「用戶端 / 伺服器」的「請求 / 回應」動態程序

其步驟如下：

1. 你在瀏覽器的網址列輸入 `http://server.com`。
2. 瀏覽器查詢 `server.com` 的 IP 位址。
3. 瀏覽器對該位址的發出一個請求，以取得網頁伺服器的首頁。
4. 這個請求經過 Internet 到達 `server.com` 網頁伺服器。
5. 收到請求的網頁伺服器從它的硬碟讀取首頁。
6. 首頁被讀入記憶體，網頁伺服器發現它是一個包含 PHP 腳本的檔案，因此將網頁傳給 PHP 解譯程式。
7. PHP 解譯程式執行 PHP 碼。
8. 有一些 PHP 有 SQL 陳述式，PHP 解譯程式將它們傳給 MySQL 資料庫引擎。
9. MySQL 資料庫將陳述式的結果回傳給 PHP 解譯程式。
10. PHP 解譯程式將 PHP 碼的執行結果以及 MySQL 資料庫回傳的結果回傳給網頁伺服器。
11. 網頁伺服器將網頁傳給發出請求的用戶端，用戶端將它顯示出來。

雖然知道這個程序以及了解三種元素如何合作很有幫助，但是在實務上，你不需要特別注意這些細節，因為它們都會自動發生。

在每一個案例中回傳給瀏覽器的 HTML 網頁也可能有 JavaScript，用戶端會在本地解譯 JavaScript，也許會啟動另一個請求，如同處理圖像等內嵌物件時的行為。

使用 PHP、MySQL、JavaScript、CSS 與 HTML5 的好處

本章的開頭介紹了 Web 1.0 的世界，但在它不久之後，Web 就進入 1.1 時代了，瀏覽器加入 Java、JavaScript、JScript（微軟稍微修改 JavaScript 的版本）及 ActiveX。伺服器端也有所進展，可透過通用閘道介面（CGI）使用 Perl 等腳本語言（PHP 語言的替代物）以及伺服器端腳本語言，動態地將某個檔案的內容（或本地端程式的輸出）插入另一個檔案中。

塵埃落定之後，有三項主要的技術脫穎而出。雖然 Perl 仍然是熱門且擁有大批追隨者的腳本語言，但 PHP 因為其簡單以及內建 MySQL 資料庫程式連結而獲得超過兩倍的使用者。JavaScript 不僅成為動態操作階層式樣式表 (CSS) 和 HTML 的重要角色，現在甚至接手更繁重的工作，負責處理非同步通訊 (在網頁載入之後，在用戶端與伺服器之間交換資料) 的用戶端。非同步通訊可在網頁使用者不知情的情況之下，讓網頁在幕後處理資料，並將請求傳給伺服器。

PHP 與 MySQL 的互利共生性質無疑推動了它們彼此的發展，但它們最吸引開發人員的地方在哪裡？簡單來說，它們可以讓你輕鬆寫意且快速地建構網站的動態元素。MySQL 是一種快速、強大，且易用的資料庫系統，提供了所有的要素來讓網站能夠查詢資料並將資料傳給瀏覽器。當你同時使用 PHP 與 MySQL 來儲存及取出資料時，你就擁有開發社交網站所需的基本元素，並且站在 Web 2.0 的起跑點了。

一旦你將 JavaScript 與 CSS 納入工具箱，你就掌握了建構高度動態且具互動性的網站的秘方—特別是現在已經有很多精密的 JavaScript 函式框架可讓你用來提升開發 web 的速度。這些框架包括著名的 jQuery (直到最近，它依然是程式設計師使用非同步通訊功能時最常用的工具之一)，以及最近正在快速掘起、越來越受歡迎的 React JavaScript，它是受到廣泛下載和實作的框架之一，以至於自 2020 年以來，在 Indeed 徵才網站上列出的 React 研發職位是 jQuery 的兩倍多。

MariaDB：MySQL 的複製品

在 Oracle 併購 Sun Microsystems (MySQL 的擁有者) 之後，開發社群開始擔心 MySQL 再也不會完全開放原始碼了，於是從它衍生一個分支，MariaDB，在 GNU GPL 之下提供免費的選擇。有些 MySQL 的原始開發者也帶領 MariaDB 的開發，使得它與 MySQL 維持緊密的相容性。因此，你可能會發現有些伺服器用 MariaDB 來取代 MySQL，但不用擔心，無論你是使用 MySQL 還是 MariaDB，本書的所有程式都可以正常運作。無論出於何種目的，你都可以將其中一種換成另一種，而不會察覺任何差異。

無論如何，事實證明，當初的擔心應該是多餘的，因為 MySQL 仍然開放原始碼，Oracle 的收費服務只有專業的支援，以及具備額外功能的版本，例如地理複製 (geo-replication) 與自動調整 (automatic scaling)。不過，MySQL 再也不像 MariaDB 那樣，是由社群推動的產品了。因此擁有 MariaDB 這座靠山可以讓開發人員比較睡得著，也應該可以讓 MySQL 維持開放原始碼。

使用 PHP

使用 PHP 可以輕鬆地在網頁中嵌入動態行為。一旦你將網頁的副檔名命名為 `.php`，它們就可以即時使用腳本語言。從開發者的角度來看，你只要編寫下面的程式碼就可以了：

```
<?php
echo " Today is " . date("l") . ". ";
?>
Here's the latest news.
```

開頭的 `<?php` 會要求網頁伺服器用 PHP 來解譯接下來直到 `?>` 標籤為止的程式碼。在這個結構之外的所有東西都會被當成 HTML 送到用戶端。所以瀏覽器會直接輸出文字 `Here's the latest news.`。在 PHP 標籤裡面，內建的 `date` 函式會根據伺服器的系統時間來顯示目前是星期幾。

這兩個部分的最終輸出為：

```
Today is Wednesday.Here's the latest news.
```

PHP 是靈活的語言，有些人喜歡直接把 PHP 結構放在 PHP 碼的旁邊，例如：

```
Today is <?php echo date("l"); ?>.Here's the latest news.
```

此外還有其他的格式化與資訊輸出方法，我會在 PHP 的章節中說明。我在這裡想要強調的是，PHP 讓網頁開發人員擁有一種非常快速的語言（雖然不像 C 或與之類似的語言那麼快），而且它可以和 HTML 標籤無縫地整合。



如果你想要在程式編輯器中輸入本書的 PHP 範例來同步操作，務必記得在它們前面加入 `<?php`，並且在後面加入 `?>`，以確保 PHP 解譯器會處理它們。為了方便起見，你可以準備一個名為 `example.php` 的檔案，並且在裡面加入這些標籤。

使用 PHP 後，你就可以自在地控制你的網頁伺服器了，無論你需要即時修改 HTML、處理信用卡、將使用者資訊加入資料庫，或是從第三方網站抓取資訊，你都可以同一組包含 HTML 的 PHP 檔案裡面作業。

使用 MySQL

當然，除非你能夠記錄使用者在你的網站提供的資訊，否則動態改變 HTML 的輸出沒有什麼意義。在 web 早期，很多網站都會使用「平面 (flat)」文字檔來儲存帳號與密碼等資料。但是你必須正確地使用鎖定機制，來避免這種檔案被多人同時存取時損毀，否則這種方法會造成很大的麻煩。此外，一般的檔案可能會變很大，之後會臃腫得難以管理—更何況，合併檔案非常困難，在合理的時間之內執行複雜的搜尋也是如此。

這就是為什麼關聯式資料庫以及結構化查詢如此重要。由於 MySQL 是免費的，而且被廣泛地安裝 Internet 網頁伺服器上，所以非常適合用來處理這種情況。MySQL 是一種穩健且快速的資料庫管理系統，使用類似英文的指令。

MySQL 結構的最高層是資料庫，你可以在裡面加入一或多個資料表來儲存資料。例如，假設你有一個資料表，名為 *users*，在裡面，你已經建立了 *surname*、*firstname* 與 *email* 欄位，當你想要加入另一位使用者時，可以使用這個指令：

```
INSERT INTO users VALUES('Smith', 'John', 'jsmith@mysite.com');
```

當然，你必須先發出其他的指令來建立資料庫與設定所有欄位，展示這個 SQL INSERT 指令只是為了讓你知道，在資料庫裡面加入新資料有多麼簡單。SQL 是在 1970 年代早期設計的語言，該年代讓人聯想到最古老的程式語言之一：COBOL。然而，它非常適合用來查詢資料庫，這也是它過了這麼久還被使用的原因。

用它來查詢資料也很簡單。假設你有一位使用者的 email 地址，你可以發出這個 MySQL 查詢來找出那個人的名字：

```
SELECT surname,firstname FROM users WHERE email='jsmith@mysite.com';
```

MySQL 會回傳 Smith, John，以及在資料庫中，與該 email 有關的其他姓名。

你應該可以猜到，除了簡單的 INSERT 與 SELECT 指令之外，MySQL 還有其他用途。例如，你可以結合相關的資料組，來將彼此相關的資訊放在一起，或是以各種順序取出結果，或是用一部分的字串來找出完整字串，或是只回傳第 n 筆結果……等。

你可以直接用 PHP 來對 MySQL 發出以上所有呼叫，而不需要自己直接操作 MySQL 命令列介面。這意味著，你可以把結果放在陣列裡面進行處理，並執行多次查詢，其中的每一次查詢都使用前一次的查詢回傳的結果，來找出你需要的資料項目。

你之後會看到，MySQL 有一些強大的內建函式可讓你高效地執行常見的 MySQL 操作，讓你免於使用 PHP 來多次呼叫 MySQL。

使用 JavaScript

JavaScript 的目的是為了讓你使用腳本來操作 HTML 文件的所有元素；換句話說，它可以讓使用者進行動態的互動，例如在輸入表單中檢查 email 地址是否有效，以及顯示諸如「你的意思真的是那樣嗎？」的提示（但不能用它來做安全維護，這項工作通常要在網頁伺服器上進行）。

網頁顯示動態變化的做法是讓 JavaScript 與 CSS（見後續章節）在幕後運作，而不是用伺服器回傳新網頁。

然後，由於不同的瀏覽器可能用不同的方式來實作 JavaScript，所以 JavaScript 使用起來可能有點麻煩。主要的原因是，有些製造商試圖在瀏覽器中加入一些功能，因而降低了與競爭對手之間的相容性。

值得慶幸的是，現在大部分的開發人員都已經了解他們必須彼此完全相容了，所以現在你應該不需要為各式各樣的瀏覽器優化你的程式碼。然而，目前仍有數百萬用戶使用舊瀏覽器，這種情況可能還會持續多年。幸運的是，有一些方法可以解決不相容問題，本書稍後介紹的一些程式庫與技術可以讓你安全地忽略這些差異。

現在，我們來看如何使用基本的 JavaScript，所有瀏覽器都接受它：

```
<script type="text/javascript">
  document.write("Today is " + Date() );
</script>
```

這段程式要求瀏覽器將 `<script>` 標籤裡面的東西都當成 JavaScript 來解譯，瀏覽器會將文字 `Today is` 與日期（用 JavaScript 函式 `Date` 取得的）寫至目前的文件。其結果如下：

```
Today is Wed Jan 01 2025 01:23:45
```



除非你需要明確指定 JavaScript 版本，否則通常可以省略 `type="text/javascript"`，只要使用 `<script>` 來開始解譯 JavaScript 即可。

如前所述，JavaScript 最初的目的是為了動態控制 HTML 文件裡面的各種元素，這也是它目前主要的用途。但是越來越多人用 JavaScript 來做非同步通訊，也就是在幕後與伺服器進行通訊的程序。

非同步通訊可將網頁變成類似獨立的程式，因為網頁不需要為了顯示新的內容而執行重新載入，只要用一個非同步呼叫，就可以拉出並更新網頁的一個元素，例如改變社交網站上的照片，或是將你按下的按鈕換成問題的答案。第 18 章會完整地討論這個主題。

在 jQuery 簡介裡，我們會說明 jQuery 框架，當你需要使用快速、跨瀏覽器的程式碼來操作網頁時，可以用它來節省重造車輪的時間。當然，現在也有其他的框架可供使用，所以我們也會在第 24 章介紹 React，它是現今最流行的選項之一。兩者都非常可靠，是許多經驗豐富的網路開發者的主要工具。

使用 CSS

CSS 是 HTML 的重要夥伴，可確保 HTML 文字與內嵌的圖像有一致的版面，並且適合在使用者的螢幕上顯示。隨著 CSS3 標準在最近幾年的問世，現在的 CSS 可以提供一些過往只能用 JavaScript 來產生的動態互動功能。例如，你不但可以裝飾任何一種 HTML 元素，改變它的尺寸、顏色、邊框、間距……等，現在只要用幾行 CSS 就可以在網頁中加入動態的轉變效果。

若要使用 CSS，你只要在網頁標頭的 `<style>` 與 `</style>` 標籤之間插入一些規則即可，例如：

```
<style>
  p {
    text-align:justify;
    font-family:Helvetica;
  }
</style>
```

這些規則會改變 `<p>` 標籤的預設文字對齊方式，讓它裡面的段落完全對齊並使用 Helvetica 字型。

第 19 章會教你用許多方式來編排 CSS 規則，你也可以直接把它們放入標籤裡面，或是將一組規則放在外部檔案內，以後再分別載入。這些靈活的做法不但可以讓你精確地設計 HTML，也可以（例如）提供內建的暫留（hover）功能，在滑鼠經過物件時產生動畫效果。你也會學到如何從 JavaScript 與 HTML 存取元素的所有 CSS 屬性。

此外也有 HTML5

雖然網頁標準已經具備許多實用的功能了，但它們仍然無法滿足雄心勃勃的開發人員。例如，目前你必須藉助 Flash 等外掛程式才能輕鬆地操作瀏覽器中的圖形。在網頁中插入音訊與視訊也一樣。此外，在 HTML 的發展過程中，也有一些麻煩的不一致性。

為了解決這些問題，並且讓 Internet 進步到 Web 2.0，進入它的下一個週期，有一種新的 HTML 標準被建立出來，以處理這些缺點：*HTML5*。HTML5 早在 2004 年就開始開發了，當時它的第一份草案是 Mozilla Foundation 與 Opera Software（兩種熱門瀏覽器的開發者）擬定的。但是一直到 2013 年初，在最終的草案被送到 World Wide Web Consortium（W3C，網頁標準的國際理事機構）之後，它才正式起步。

HTML5 花了好幾年的時間開發，但是現在我們有非常穩健的 5.1 版（2016 年起）。不過它的研發週期是永無止盡的，隨著時間的過去，一定會有更多功能被加入，第 5.2 版（打算淘汰外掛系統）在 2017 年發表，它是 W3C 推薦版本，HTML 5.3（具有自動首字母大寫等建議功能）在 2020 年仍然在規劃中。HTML5 有一些處理與顯示媒體的優秀功能，包括 `<audio>`、`<video>` 與 `<canvas>` 元素，分別可加入音訊、視訊與進階圖片。第 25 章會詳細介紹它們與 HTML 5 的其他層面。



HTML5 規格有一個我很喜歡的小地方：XHTML 語法再也不需要 self-closing 元素了。在之前，你可以使用 `
` 元素來顯示一個分行符號。之後，為了確保未來與 XHTML（原本打算取代 HTML，卻從未實現的一種語言）相容，它被改為 `
`，加入結尾的字元 /（因為所有元素都應該有一個包含這個字元的結束標籤）。但是現在事情圓滿解決了，你可以使用這個元素的任何一種版本。因此，為了保持簡潔並減少打字數量，本書採用之前的風格，使用 `
`、`<hr>` 等樣式。

Apache 網頁伺服器

除了 PHP、MySQL、JavaScript、CSS 及 HTML5，動態網頁還有第六位英雄：網頁伺服器。在本書中，它是 Apache 網頁伺服器。我們已經在 HTTP 伺服器 / 用戶端的交換程序中稍微討論網頁伺服器的功能了，其實它在幕後做的事情還有很多。

例如，Apache 並非只傳送 HTML 檔，它也處理許多檔案，包括圖像、Flash 檔、MP3 音訊檔、RSS（Really Simple Syndication，真正簡易新聞訂閱方式）來源……等。但是

這些物件不一定是 GIF 圖像那種靜態檔案，它們也可能是 PHP 腳本等程式產生的。沒錯：PHP 甚至可以創造圖像及其他檔案，無論是即時創造，還是先做好再傳遞。

為此，你通常要先將模組編譯到 Apache 或 PHP 裡面，或是在執行期呼叫它們，GD（Graphics Draw）程式庫是其中一種模組，PHP 可用它來創造及處理圖形。

Apache 也支援大量的自有模組。除了 PHP 模組之外，對身為網頁程式設計師的你來說，最重要的模組是處理安全的模組。其他的模組包括 Rewrite 模組，它可以讓網頁伺服器處理各種 URL 類型，並根據內部的需求將它們重寫，以及 Proxy 模組，它可以讓你從快取將經常被請求的網頁送出，以減緩伺服器的負擔。

在本書稍後，你會看到如何使用以上的一些模組來加強三種核心技術的功能。

處理行動設備

我們已經深陷一個以行動設備來互相連接的世界，只為桌機開發網站的想法已經變得相當落伍了。現在的開發者都想開發可以根據所處環境修正行為的響應式網站與 web app。

所以，在這個版本的新內容中，我要告訴你如何輕鬆地使用本書介紹的技術來創造這類的產品，以及提供響應式 JavaScript 功能的 jQuery Mobile 程式庫。學會它們之後，你就可以把焦點放在網站與 web app 的內容與易用性上，因為它們可以根據各種類型的設備自動優化畫面，幫你免除一項煩惱。

為了展示如何完全掌握這項能力，本書的最後一章會建立一個簡單的社交網站範例，它使用 jQuery 來產生完整的響應功能，確保它可在任何設備上（包括手機螢幕、平板與桌機）顯示合適的畫面。雖然我們也可以使用 React（或其他的 JavaScript 程式庫或框架），但是當你看完這本書之後，也許可以將使用它們當成一個給自己的練習。

關於開放原始碼

本書討論的技術都是開放原始碼的，任何人都可以閱讀與修改程式。人們經常爭論開放原始碼是不是讓這些技術如此流行的原因，但無論如何，PHP、MySQL 與 Apache 都是在它們各自的領域中最常見的工具。我們可以肯定地說，開放原始碼意味著那些技術是社群的程式團隊根據他們自己的需求開發出來的，而且提供原始的程式碼，讓所有人都可以閱讀與修改，所以 bug 會被快速發現，而且安全漏洞會在問題發生之前被解決。

此外，開放原始碼還有一項優點：這些程式都是免費使用的。如果你想要擴充網站並增加伺服器，不用擔心要不要購買額外的授權。當你想要將這種產品升級成最新版本時，也不需要計算預算還有多少。

整合

PHP、MySQL、JavaScript（有時會藉助 React 或其他的框架）、CSS 及 HTML5 是製作動態網頁內容的絕佳組合：PHP 可在網頁伺服器處理所有主要工作，MySQL 可管理所有資料，而 CSS 與 JavaScript 的組合可處理網頁的外觀。當 JavaScript 需要更新某些東西時（無論是在伺服器或網頁上），它也可以與網頁伺服器上的 PHP 程式溝通。同時，因為 HTML5 強大的新功能，例如 canvas、音訊與視訊以及地理定位，你可以讓網頁產生高度的動態、互動性與多媒體效果。

我們來總結本章的內容，在不展示程式碼的情況下，介紹如何結合這些技術來產生常見的非同步通訊功能：當使用者註冊新帳號時，檢查他輸入的帳號是不是被網站的其他人建立了。Gmail 是這種功能的例子（見圖 1-3）。

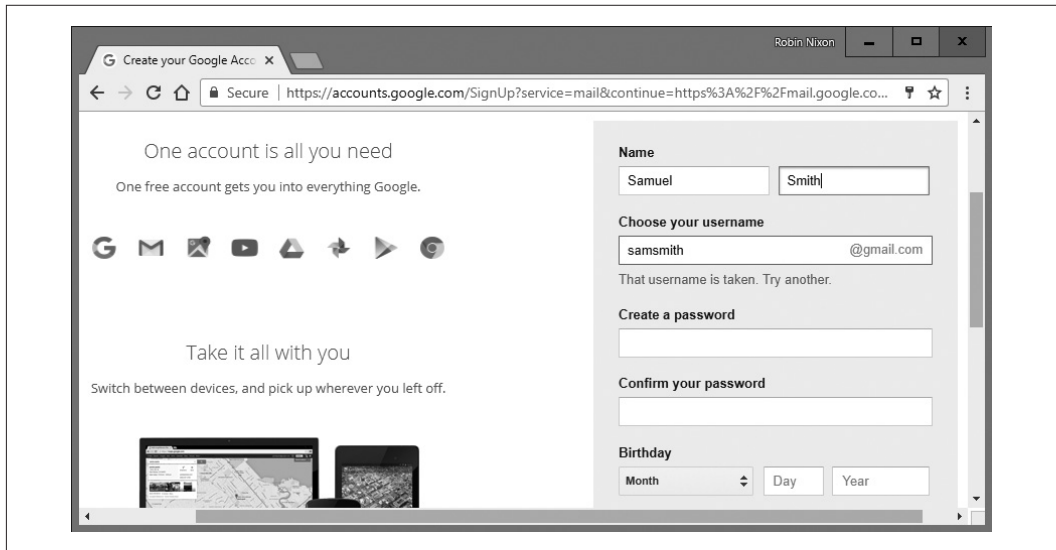


圖 1-3 Gmail 使用非同步通訊來檢查帳號是否可用

這個非同步程序的步驟類似這樣：

1. 伺服器輸出 **HTML** 來建構網頁表單，索取必要的資料，例如帳號、姓、名，和 **email** 地址。
2. 同時，伺服器會在 **HTML** 附加一些 **JavaScript** 來監視帳號輸入欄位並檢查兩件事：(a) 使用者是否在裡面輸入文字，及 (b) 使用者是否按下其他的輸入方塊而選取輸入欄位之外的欄位。
3. 當使用者輸入文字，而且該欄位不再被選取時，**JavaScript** 碼在私底下將已輸入的帳號回傳給伺服器的 **PHP** 腳本，並等候回應。
4. 網頁伺服器查看帳號，將該帳號是否已被使用的結果回傳給 **JavaScript**。
5. **JavaScript** 在帳號輸入方塊旁邊顯示文字，讓使用者知道帳號是否可用 —— 也許是用綠色的打勾記號或紅色的打叉圖案，並且附上一些文字。
6. 如果該帳號無法使用，但使用者仍然將表單送出去，**JavaScript** 會中斷提交動作，並再次強調（也許使用比較大張的圖片及 / 或警示方塊）使用者必須選擇其他的帳號。
7. 這個程序的進階版本甚至可以根據使用者提供的帳號來建議目前可用的其他選項（選擇性的功能）。

這些步驟都在幕後快速地執行，產生舒適且無縫的使用者體驗。如果不使用非同步通訊，瀏覽器就必須將整個表單送給伺服器，再由伺服器回傳 **HTML**，並標出所有的錯誤，雖然這種做法也無不可，但是它遠不如即時處理表單欄位那樣明快且舒適。

非同步通訊並非只能處理簡單的輸入驗證與處理，本書稍後會介紹更多可以用它來處理的事情。

本章詳細介紹了 **PHP**、**MySQL**、**JavaScript**、**CSS** 與 **HTML5**（及 **Apache**），並且說明它們如何互相合作。第 2 章會帶著你安裝網頁開發伺服器，讓你可在上面做練習。

問題

1. 建構完全動態的網頁必備的元件（至少）有哪四個？
2. *HTML* 是什麼意思？
3. 為什麼在 *MySQL* 這個名稱裡面有 *SQL* 這三個字母？
4. **PHP** 與 **JavaScript** 都是在網頁產生動態結果的程式語言，它們的主要差異為何？為什麼要同時使用兩者？
5. *CSS* 是什麼意思？
6. 列出 **HTML5** 加入的三個主要新元素。
7. 如果你在開放原始碼工具中遇到 **bug**（這很罕見），如何修正它？
8. 為什麼 **jQuery** 或 **React** 這類的框架對現代網站與 **web app** 的開發非常重要？

解答請參考第 763 頁，附錄 A 的「第 1 章解答」。