

---

# 前言

JavaScript 原本計劃作為客戶端載入網頁的瀏覽器（當時為 Netscape Navigator），和伺服器端應用程式之間的腳本（scripting）介面。自 1995 年發布至今，JavaScript 已變成了開發 Web 的重要元件，而且也發現到處都在使用。

本書涵蓋的 JavaScript 語言，從最基本的資料型態—從這個語言一開始就跟隨著的觀念，到最複雜的功能—包括那些涉及 Ajax 和 DHTML 的部分。讀完這本書時，你將擁有所需的基礎知識，甚至能處理最複雜的函式庫和 Web 應用程式。

## 本書讀者對象

本書的讀者應當熟悉網頁技術，包括 CSS 和 HTML/XHTML，還應該見過一些網頁中的 JavaScript。此外，並不需要有程式設計經驗，然而如果沒有程式設計經驗，可能需要額外地多看幾次某些章節才能看懂。

本書應該對下面人士有所助益：

- 任何想要或需要把 JavaScript 整合到自己個人網頁或網站的人。
- 任何使用內容管理工具（比如部落格工具）並更想理解納入到工具樣板中的腳本元件的人。
- 尋求將 JavaScript 與某些 DHTML/Ajax 特性，整合到其網站的 Web 開發人員。
- 想為客戶開發新市場的 Web 服務開發人員。
- 講授 Web 技術相關內容的教師。
- 希望深入理解如何使用互動或動畫效果，使自己的設計變得更加生動的網頁設計師。

- 任何對 Web 技術感興趣的人。

## 使用本書

正如前面說過的，本書假設讀者已有 (X)HTML 和 CSS 方面的經驗，且對 Web 應用程式的運行原理有所瞭解。程式設計經驗不是必需的，但是本書涵蓋 JavaScript 的所有面向，其中有些內容相對複雜一些。雖然較難的章節很少，但還是需要讀者對 JavaScript 有足夠的瞭解才能使用較新的 Ajax 函式庫。

本書分為 4 部分：

第 1 章至第 3 章介紹 JavaScript 應用的結構，包括該語言所支援的簡單資料型態，還有基本敘述和控制結構。這為在後續的章節中理解 JavaScript 打下了基礎。

第 4 章至第 8 章介紹主要的 JavaScript 物件，包括所有重要的函式、網頁表單的腳本存取、事件處理、腳本安全性和使用 cookie。這些主題組合在一起就構成 JavaScript 的核心。讀完這些章節之後，讀者就可以驗證表單元素、設定和擷取 cookie、捕捉事件並為事件提供功能，甚至還能建立 JavaScript 函式庫。這幾章所講述的功能近 10 年來一直都是 JavaScript 的基礎，並且至少在未來 10 年中還是這樣。

第 9 章到第 11 章將深入網頁開發更複雜的面向。這幾章涵蓋瀏覽器物件模型 (Browser Object Model) 和較新的文件物件模型 (Document Object Model, 之後簡稱 DOM)，並展示如何自訂自己的物件。如果你想建立一個新視窗，或者個別存取、修改甚至動態地建立任何頁面元素，理解這些模型就非常重要。另外，有了自訂物件，就可以超越語言或瀏覽器本身所預置的那些能力。

第 12 章到第 14 章將進入到 JavaScript 的高級應用，包括 DHTML、Ajax 以及對兩者都支援的眾多精彩的新函式庫。

### 第 1 章，初探 JavaScript

介紹 JavaScript 並給出一個小型的網頁應用進行快速的初體驗。本章還講述了一些與使用 JavaScript 相關的議題，包括很多可用的工具，還有安全性和可存取性的議題。

### 第 2 章，JavaScript 的資料型態與變數

提供了 JavaScript 的基本資料型態的概覽，以及對於語言變數、識別字和 JavaScript 敘述結構的概述。

### 第 3 章，運算子和敘述

講述了 JavaScript 的基本敘述，包括指定敘述、條件敘述和控制敘述，還有這三種敘述所必需的運算子。

## 第 4 章，JavaScript 物件

介紹了一些首要的 JavaScript 內建物件，包括 `Number`、`String`、`Boolean`、`Date` 和 `Math`。本章還介紹了 `RegExp` 物件，它提供了正規運算式模式比對的功能。在檢查表單欄位時，正規運算式很重要。

## 第 5 章，函式

專注在另一個 JavaScript 內建物件：函式 (function)。函式是建立自訂物件的關鍵，它還能將 JavaScript 程式碼區塊封裝成一段段，以便在不同的 JavaScript 應用中重覆使用。JavaScript 函式相對簡單，但在某方面也可能很複雜，其中包括遞迴 (recursion) 和 closure，本章也介紹這兩個議題，細節則涵蓋在第 11 章。

## 第 6 章，捕捉事件

專注在事件處理，包括原始的事件處理形式 (依然廣泛使用於許多應用中)，還有較新的基於 DOM 的事件處理形式。

## 第 7 章，表單與即時驗證

介紹 JavaScript 在表單和表單欄位上的使用，包括如何存取各種欄位型態 (例如文字輸入欄位和下拉清單) 以及驗證擷取的資料。在表單提交給 Web 伺服器之前進行表單驗證，以避免與伺服器之間有不必要的往返資料傳輸，因而相互節省時間和資源的使用。

## 第 8 章，沙箱及其他的議題：cookie、互通性和隱私

講述以腳本為基底的 cookie (它在客戶機器上存儲了小塊的資料)。有了 cookie，可以存儲使用者名稱、密碼和其他資訊，這樣使用者就不必總是重新輸入資料。此外，由於討論 cookie 不可避免地要引發對安全性的討論，本節還講述了一些與 JavaScript 相關的安全問題。

## 第 9 章，基礎瀏覽器物件

開始介紹從 JavaScript 可存取到的物件模型。首先是**瀏覽器物件模型 (Browser Object Model, BOM)**，它是物件的層次結構，其中的物件包括 `window`、`document`、`forms`、`history` 和 `location` 等等。透過 BOM，JavaScript 可以打開視窗，存取 `forms`、`links` 和 `images` 之類的頁面元素，甚至實現一些基本的動態效果。

## 第 10 章，DOM：文件物件模型

專注於**文件物件模型 (Document Object Model, DOM)**，DOM 是一個簡單而且舉足輕重的物件模型，它提供了對於所有文件元素和特性的存取。你將看到基於 XML (比如 XHTML) 和 HTML 的文件。雖然模型容易理解並且作用範圍也相當明瞭，但對於程式設計新手來說，本章的某些內容還是具有一些挑戰性。

### 第 11 章，建立自訂的 JavaScript 物件

展示如何在 JavaScript 中建立自訂物件，講述此語言中用以建立這類自訂結構的所有原型結構。討論了一些程式設計語言的概念，比如繼承和封裝，但你並不需要有這些概念的經驗。

### 第 12 章，構建動態網頁：在腳本中加入樣式

概括性地介紹一些較常用的動態 HTML 效果，包括拖放操作、折疊 / 展開一部分頁面內容、可見性效果和移動效果。閱讀本章需要對 CSS 有一定程度的瞭解。

### 第 13 章，使用 Ajax

本章介紹 Ajax。無論產生怎樣令人激動的效果，Ajax 實際上並不是 JavaScript 的複雜運用。除了講述 Ajax 的元件之外，本章還提供了一個 Google Maps 的例子。在對於 Ajax 的推動上，這個應用可能比任何其他應用都要大。

### 第 14 章，好消息：豐腴的函式庫！令人驚艷的 Web 服務！有趣的 API！

講述一些較流行、可以免費下載和使用的函式庫。其中包括 Prototype、Sabre's Rico、Dojo、MochiKit、Yahoo! UI 和 script.aculo.us。透過這些函式庫和本書，你將擁有所需的技能建立出難以置信的、有用的 Web 應用。

## 本書印刷體裁

本書使用下列排版體裁：

### 粗體字

強調語氣，或是標示初次出現的術語。

### 斜體字 (*Italic*)

URL、Email 地址、檔案名和文件副檔名。

### 定寬體字 (`Constant width`)

表明是廣義的電腦程式碼，包括命令、陣列、元素、敘述、選項、分支、變數、元素特性、關鍵字、函式、類型、類別、名稱空間、方法、模組、屬性、參數、值、物件、事件、事件控制碼、XML 標記、HTML 標記、巨集、檔案的內容和命令的輸出結果。

### 定寬粗體 (**Constant bold**)

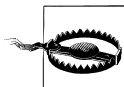
表示應由用戶逐字輸入的命令或其他文字。

### 等寬斜體 (*Constant width italic*)

表示應由用戶所提供的值或根據上下文環境所確定的值進行替換的文字。



表示提示、建議、或一般注解。



表示警告或注意事項。

本書所提到的網站和網頁能幫你找到一些可能有用的線上資訊。通常，我都指出了這些頁面的網址（URL）和名稱（標題）。有些網址相對複雜一些，但透過你自己喜歡的搜尋引擎，根據網頁的名稱搜尋，可能更容易找到這些網頁，通常應把名稱寫在引號中。如果根據書中給的網址找不到對應網頁，搜尋名稱也對你有所幫助；有可能網頁被移到別的地方去，但名稱可能仍然有作用。

## 程式範例的使用

這本書是協助你把工作做好。一般而言，你可以在你的程式和說明文件中使用本書的程式碼。除非你要重製重要的程式碼，否則無需取得我們的許可。例如，使用本書片片斷斷的程式碼寫了一個程式，並不需要取得我們的許可。但是，把 O' Reilly 書籍的程式範例燒成光碟片販賣或散佈，就需要取得授權。引用本書的文句和範例程式碼回答問題，不需要取得許可。把本書大量的程式範例整合到你產品的說明文件中，則需要取得授權。

雖然不是必要，但註明來源，我們會很感謝。註明來源通常包括書名、作者、出版商、以及 ISBN。例如，「Learning JavaScript by Shelley Powers. Copyright 2007 O' Reilly Media, Inc., 978-0-596-52746-4」

如果你覺得你對書中程式範例的使用情況有別於上述情況，不用客氣，盡量和我們連絡：permissions@oreilly.com。

## 連絡我們

本書有網頁，列出錯誤、範例、或其他資訊（如果有的話），網址在：

英文網頁：

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

中文網頁：

[http://www.oreilly.com.tw/product2\\_web.php?id=a226](http://www.oreilly.com.tw/product2_web.php?id=a226)

也可以造訪本書作者的網站：

<http://learningjavascript.info>

要提供本書之意見或詢問技術問題，可以把信寄到：

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

有關我們的書籍、會議、Resource Center、O'Reilly Network 的資訊，可以看一看我們的網站：

英文網：

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

中文網：

<http://www.oreilly.com.tw/>

## 誌謝

對於有些書，在你的身後有一個非常好的團隊，本書就是這樣。我想感謝我的編輯 Simon St.Laurent，感謝他的耐心、熱情和在本書寫作過程中出現偏差時所給予的指導。另外，我要感謝技術及內容的審校人員 Steven Champeon、Roy Owens 和 Alan Herrell，感謝他們出色的建議和在發現漏洞和不當方面給予的幫助。

我還要感謝本書的製作編輯 Rachel Monaghan、文字編輯 Mary Anne Weeks Mayo、索引製作人 Johnna VanHoose Dinse 和製作經理 Marlowe Shaeffer。

最後，我要感謝那些在網路上碰到的人—有的是在技術社群中，有的不是。在我寫這本書時，我記得你們。在某種程度上，可以說這本書是為你們而寫的，你們知道我說的是誰。