

# 序

這本書是「C 編程語言」和「C 執行期程式庫」的完整參考。既然是一本 Nutshell 書籍，這本書的目的是提供一本方便、可靠的手冊，對你日常的編程工作有幫助。本書描述所有的語言元素，並展示它們的用法。

本書對於 C 語言的描述，是根據 1999 年的國際 C 標準「ISO/IEC 9899:1999」，也包含了技術修正「2001 年的 TC1」和「2004 年的 TC2」。此標準被稱為 C99，是「ISO/IEC 9899:1990 標準」和「1995 基準增補 1 (ISO/IEC 9899/AMD1:1995)」的擴充。1990 ISO/IEC 標準對應於 ANSI 標準 X3.159，是在 1989 年底核可的標準，常常被稱為 ANSI C 或 C89。

並非所有的編譯器和標準程式庫實踐版本都已經完全支援新的 1999 C 標準的諸多特色。本書因此會對於 1999 的擴充部分（像是早期版本沒有的函式），特別標上「C99」的註記。

本書並非 C 程式設計的入門書。雖然本書有涵蓋語言的基礎部分，但是本書的編排和組織方式並不適合初學者。如果你是 C 的初學者，我們假設你至少要讀過任何一本 C 的入門書，或者至少熟悉相關的語言（例如 Java 或 C++），才適合閱讀本書。

## 本書的組織方式

本書分成三部分。第一部份描述 C 語言，用的是相當嚴謹的詞彙；第二部分描述標準程式庫；第三部分描述編譯和測試，搭配 GNU 軟體中常用的工具。

## 第一部份

第一部份 (Part I) 處理 C 語言，包括第 1 章到第 14 章。在第 1 章之後，描述語言的一般概念和元素，每章都分配到一個主題，例如型別、敘述、指標。主題出現的次序有依據基本的概念來安排，讓後面的主題會以前面的主題當基礎。比方說，先介紹型別，然後才會介紹表示式和運算子，然後才是敘述。你有時候會需要參考到後面的章節，來填補一些相關的細節。比方說，在第 5 章提前做一些指標和陣列的討論，是有必要的 (第 5 章涵蓋表示式和運算子)，不過詳細的指標和陣列，要看第 8 和第 9 章。

### 第 1 章，語言基礎

描述語言的特性，以及 C 程式如何架構和編譯。本章介紹一些基本概念，像是翻譯單元、字元集、識別字。

### 第 2 章，型別

提供 C 型別的整體概觀，並描述基本型別、void 型別、列舉型別。

### 第 3 章，字面值

描述數字常數、字元常數、字串字面值 (包括逃避序列)。

### 第 4 章，型別轉換

描述顯性和隱性的型別轉換，包括整數提升和通常的算術轉換。

### 第 5 章，表示式與運算子

描述表示式的估算、所有的運算子、以及和運算子相容的運算元。

### 第 6 章，敘述

描述 C 語言的敘述，像是區塊、迴圈、跳躍。

### 第 7 章，函式

描述函式定義，和函式呼叫，包括遞迴與 inline 函式。

### 第 8 章，陣列

描述固定寬度和變動寬度的陣列，包含字串、陣列初始化、多維度陣列。

### 第 9 章，指標

描述物件指標和函式指標的定義和使用。

### 第 10 章，結構、聯合、位元欄位

描述在「使用者定義的資料型別」內的資料組織方式。

### 第 11 章，宣告

描述宣告的一般語法、識別字連結、物件儲存期限。

### 第 12 章，動態記憶體管理

描述標準程式庫的動態記憶體管理函式，實踐一個二元樹以展示這些函式的使用方式。

### 第 13 章，輸入和輸出

描述輸入和輸出的概念，以及標準 I/O 程式庫的使用方式概觀。

### 第 14 章，前處理假指令

描述巨集的定義和使用、條件式編譯、以及所有其他的前處理假指令和操作。

## 第二部分

第二部分包含第 15 章到第 17 章，內容是 C 的標準程式庫。提供了標準標頭概觀，以及函式使用上的細節。

### 第 15 章，標準標頭

描述標準標頭和用途。包含所有標準程式庫的巨集和型別定義。

### 第 16 章，函式概觀

提供標準函式庫函式的整體概觀，根據應用的方式組織，像是「數學函式」、「時間和日期函式」等。

### 第 17 章，標準程式庫函式

詳細描述每個標準程式庫函式，以字母順序排列，用範例展示每個函式的用法。

## 第三部分

本書第三部分，提供 C 編程員基本工具的必要知識：編譯器、make 工具程式、除錯器。這裡所描述的工具都是在 GNU 軟體系列中。

### 第 18 章，使用 GCC 進行編譯

描述廣為使用的 GCC 編譯器對 C 編程員提供哪些能力。

### 第 19 章，使用 make 建構 C 程式

描述如何使用 *make* 程式來讓大型程式的編譯程序自動化。

## 第 20 章，利用 GDB 為 C 程式除錯

描述如何「在 GNU 除錯器控制之下」執行程式，以及如何分析程式的執行時期行為以找出邏輯錯誤。

## 進階閱讀

書中許多地方都有提及更進一步的參考資料，除此之外，還有一些資源提供更多細節。國際 C 標準的官方網站在 <http://www.open-std.org/jtc1/sc22/wg14>，裡面有連結指到最新版的 C99 標準，還有目前該工作小組進行中的計畫。

如果讀者不只對於 C 的 what 和 how 感興趣，也想知道 why，WG14 的網站也提供了一個連結「C99 Rationale」解釋許多設計上的源由：這份文件並非規範，而是描述標準化進程中的動機和限制。C89 Rationale 則是放在 <http://www.lysator.liu.se/c/rat/title.html>。如果你想知道 C 如何「一開始變成那樣的」，你不可錯過 C 原創者 Dennis Ritchie 的一篇文章「The Development of the C Language」（C 語言的孕育），還有他在貝爾實驗室時的其他歷史文獻，網址在 <http://cm.bell-labs.com/cm/cs/who/dmr>。

如果你想要知道浮點數學細節的資訊，這超出 C 的範疇，你可以閱讀 David Goldberg 的完整介紹，「What Every Computer Scientist Should Know About Floating-Point Arithmetic」，目前該文件的網址在 [http://docs.sun.com/source/806-3568/ncg\\_goldberg.html](http://docs.sun.com/source/806-3568/ncg_goldberg.html)。

## 字型體裁

本書使用以下的字型體裁：

斜體字 (*Italic*)

用於目錄與檔名。範例中註解需強調的部分也會以斜體字表示。

粗體字 (**Bold**)

使用在表示 C 關鍵字，以及第一次提及的新名詞和概念。

定寬字 (`constant width`)

使用在表示程式和程式的元素，以及範例中顯示檔案的內容或命令的輸出。在正文中引用範例或程式片段的單詞或項目也以定寬字表示。

### 定寬粗體字 (**constant bold**)

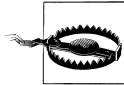
在範例中用來顯示命令或者其他需要由使用者逐字輸入的文字。

### 定寬斜體字 (*Constant italic*)

用在例子中表示應視情況取代的變數（例如，變數 *filename* 應該用某個實際的檔名來代替）。



此圖表示小技巧、建議、注意事項。



此圖表示要特別小心的地方。

## 批評和建議

雖然我們已盡最大可能測試驗證過本書中的資訊，但你可能發現到書中所描述某種功能實際上已經改變了，或甚至根本是我們搞錯了。請將你所發現的錯誤，還有你對本書未來版本的任何寶貴意見告訴我們。來信請寄：

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

網址：<http://www.oreilly.com.tw>

電子郵件：

[sales@oreilly.com.tw](mailto:sales@oreilly.com.tw)（業務部）

[editors@oreilly.com.tw](mailto:editors@oreilly.com.tw)（編輯部）

[bookquestion@oreilly.com.tw](mailto:bookquestion@oreilly.com.tw)（疑難雜症）

請以電子郵件的方式與我們聯絡，這會比電話和舊有的郵件方便。如果你買到的書有印刷品質上的問題，可以寫信到業務部；若你對書籍內容有疑義，或是發現錯字，請寫信到 [bookquestion@oreilly.com.tw](mailto:bookquestion@oreilly.com.tw)。

O'Reilly 的每一本書都有專屬網頁，你可以在此找到關於本書籍的相關資訊，包括範例程式的下載、勘誤表與相關資源的連結。

*<http://www.oreilly.com/catalog/cinanut/>*

（本書英文版的網頁）

*[http://www.oreilly.com.tw/product2\\_c.php?id=a205](http://www.oreilly.com.tw/product2_c.php?id=a205)*

（本書中文版的網頁）