

cw TEX 排版系統

“[T]he \TeX research project that I embarked on was driven by two major goals. The first goal was *quality*: we wanted to produce documents that were not just nice, but actually the best.”

“I never intended to have a system that would be universal and used by everybody. I always wanted to write a system that would be used for just the finest books.”

“The current version number for \TeX is 3.1, and for \METAFONT it is 2.7. If corrections are necessary, the next versions of \TeX will be 3.14, 3.141, then 3.14159, ..., converging to the ratio of a circle’s circumference to its diameter; for \METAFONT the sequence will be 2.71, 2.718, ..., converging to the base of natural logarithms. I intend to be fully responsible for all changes to these systems for the rest of my life.”

Donald E. Knuth
Digital Typography (1999)

排 cwTeX

版

修訂 2 版

系

吳聰敏。吳聰慧

統

版權聲明:

- 本書所提及的商標, 均屬於其合法註冊公司所有。
- cwTeX 光碟內含中文處理軟體, 23 套橫排中文字型檔與 5 套直排中文字型檔。以上軟體之版權屬吳聰敏與吳聰慧所有。上列軟體中, 中文處理軟體與 5 套橫排中文字型檔置於 cwTeX 網站與複製 (mirror) 網站上, 可供免費下載使用。網站上軟體可自由流通, 但不能修改。欲流通網站上軟體與字型檔者, 請依照 ETEX Project Public License 之條件。詳細說明, 請見 CTAN: macros/latex/base/lppl.txt 。

©2002

出版者: 吳聰敏·吳聰慧

封面設計: 吳聰敏

內文設計/排版: 吳聰敏

本書使用字體包括:

cwTeX 中文字體

Adobe Minion

Computer Modern typewriter

Y&Y Mathtime plus

總經銷:

翰蘆圖書出版有限公司

台北市懷寧街92號5樓

電話: (02)2382-2333

傳真: (02)2388-6655

郵撥: 15718419

Email: hanlu@hanlu.com.tw

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

cwTeX 網址:

<http://ceiba.cc.ntu.edu.tw/tmwu>

中文 TeX 之 BBS 討論站:

<telnet://140.112.18.32>

1997年10月初版

2000年3月2版

2002年4月修訂2版

定價: 新台幣 600 元

cwTeX 排版系統 / 吳聰敏。吳聰慧 [著]. — 修訂 2 版. —

臺北市: 吳聰敏出版: 翰蘆圖書總經銷, 2002 [民 91]

面; 公分

參考書目: 面

ISBN 986-99950-7-1 (平裝附光碟片)

1.TEX (電腦程式)

312.949T46

910003997

目錄

修訂版重要更新	15
2 版序	17
1 前言	19
2 例子	23
3 排版方法	39
3.1 排版步驟	40
3.2 WinEdt 功能簡介	45
3.3 排版訊息	46
4 安裝與設定	49
4.1 準備安裝	49
4.1.1 取得軟體	50
4.1.2 移除舊版	51
4.2 安裝工具程式	52
4.2.1 WinEdt	52
4.2.2 Ghostscript 與 GSview	52
4.2.3 Acrobat Reader	53
4.2.4 Windows Commander	53
4.3 安裝 MiKTeX 與 cwTeX	53
4.3.1 MiKTeX	53
4.3.2 cwTeX	54
4.4 測試與調整	56
4.4.1 調整顯示密度	56

4.4.2	選擇印表機	57
4.4.3	檔案搜尋	57
4.5	常見的安裝問題	58
4.6	其他軟體	61
4.6.1	fpTeX 系統	61
4.6.2	檔案搜尋	62
4.7	Linux 版本安裝方法	62
4.7.1	安裝程式及字型	63
4.7.2	測試	63
4.7.3	Linux 與 Windows 版本之差異	64
4.8	其他作業系統之版本	64
5	指令與文字	65
5.1	符號與註銷指令	65
5.2	英文稿件輸入原則	68
5.3	中文稿件輸入原則	70
5.3.1	標點符號	71
5.3.2	英文引號與破折號	73
5.4	排版其他西方國家文字	75
6	選用字體	77
6.1	字型規格	77
6.2	選用英數字體	78
6.2.1	字體相對大小指令	81
6.2.2	選擇字級與行距	82
6.2.3	選用任意點數之英數字體	85
6.3	選用中文字體	85
6.3.1	新式字體選用法	86
6.3.2	舊式中文字體指令	88
6.3.3	標題與特定段落之中文字體	88
6.3.4	調整中文字距	90
6.4	設定行距	90

6.4.1	選用什麼字體?	91
6.4.2	避頭點	93
7	文稿結構	95
7.1	長度與寬度	96
7.1.1	長度單位	97
7.1.2	設定版面大小	98
7.2	文稿結構與文件類別	101
7.2.1	指令選項	102
7.2.2	巨集套件	103
7.2.3	article 文件類別	104
7.2.4	研究生論文	107
7.2.5	book 文件類別	110
7.3	文稿題目	111
7.4	摘要	113
7.5	章節標題	114
7.5.1	章節標題之層級	114
7.5.2	設定章節標題之字體	116
7.5.3	titlesec 巨集套件	117
7.5.4	中文標題之字體	119
7.6	頁碼與頁面格式	120
7.6.1	頁碼	120
7.6.2	換頁	122
7.6.3	頁眉與頁足	123
7.7	目錄	124
7.8	附錄	126
8	段落編排	127
8.1	段落格式	127
8.1.1	居中與靠邊	128
8.2	調整間距	130
8.2.1	插入空白	130

8.2.2	填入細點或直線	132
8.3	引文與詩詞	132
8.3.1	quote 指令環境	133
8.3.2	verse 指令環境	133
8.4	條列指令環境	134
8.5	迷你版面指令環境	137
8.5.1	迷你版面	137
8.5.2	文字方塊	140
8.5.3	線條方塊	142
8.5.4	儲存方塊	142
8.6	註解與邊註	143
8.6.1	註解	143
8.6.2	註解編號	143
8.6.3	邊註	145
8.7	照列原文	146
8.8	交叉引述	149
8.9	comment 巨集指令	150
8.10	多欄位版面	150
9	數學式子	153
9.1	科技文稿之排版規範	153
9.2	數學環境	154
9.2.1	數學文稿輸入原則	155
9.2.2	運算符號與上下標	156
9.2.3	分式	156
9.2.4	開根號	157
9.3	數學符號	158
9.3.1	希臘字母符號	158
9.3.2	粗體數學符號	159
9.3.3	函數符號	160
9.3.4	積分與加總函數	160

9.3.5	箭號與相對關係符號	161
9.3.6	數學重音符號	168
9.3.7	上下重疊符號	169
9.3.8	連續點	170
9.4	定義與定理	171
9.4.1	amsthm 巨集套件	172
9.5	矩陣與行列式	175
9.6	amsmath 巨集套件	179
9.7	細節調整與數式編號	182
9.7.1	調整符號間距	182
9.7.2	調整符號大小	183
9.7.3	數學式居中與靠左	183
9.7.4	交叉引述	184
10	表格	185
10.1	tabbing 指令環境	187
10.1.1	以樣本行設定距離	188
10.1.2	其他控制指令	189
10.2	array 巨集套件	190
10.2.1	tabular 指令環境	190
10.2.2	控制欄位間距	192
10.2.3	booktabs 巨集套件	194
10.2.4	控制中文字距	195
10.2.5	表格內的文字段落	198
10.2.6	其他控制與設定指令	199
10.3	tabularx 巨集套件	202
10.4	表格標題與位置	204
10.4.1	浮動版面指令環境	204
10.4.2	圖表標題	205
10.5	引述表格	207
10.6	表格排版細節調整	209

10.6.1	表格小數點對齊	209
10.6.2	橫列文字對齊	211
10.6.3	表格註解	212
10.6.4	表格內加入括弧或斜線	214
10.7	彩色表格	217
10.8	超大型表格	220
10.8.1	旋轉大型表格	220
10.8.2	超長表格	220
11	引用外製圖形	227
11.1	圖形檔案規格	227
11.2	引用外製圖形	229
11.2.1	引用 PostScript 圖形	230
11.2.2	圖形中加入中文或數式	231
11.2.3	引用 PostScript 文稿檔案	234
11.3	繪製 EPS 圖形	235
11.3.1	以 PostScript 驅動程式產生 EPS 圖形檔	236
11.3.2	使用繪圖軟體繪製 EPS 圖形	237
11.3.3	引用圖形應注意之問題	240
11.4	引用描點圖形	241
11.5	圖形位置與標題	241
12	圖形與彩色	243
12.1	graphicx 巨集套件	243
12.1.1	旋轉文字圖表	244
12.1.2	放大或縮小文字圖表	245
12.2	picture 指令環境	246
12.3	彩色圖文	249
12.4	lettrine 巨集套件	251
12.5	contour 巨集套件	251
12.6	wrapfig 巨集套件	252
12.7	PSTricks 巨集套件	253

13 PostScript 字體	257
13.1 METAFONT 與 PostScript 字體	257
13.1.1 描邊字型與描點字型	258
13.1.2 PostScript 格式之 CM 字體	259
13.2 中文 PostScript 字體	259
13.2.1 重新定義中文變形字	260
13.2.2 字型對應檔	261
13.3 英文 PostScript 字體	263
13.3.1 txfonts 巨集套件	264
13.4 創造英文 PostScript 字體巨集套件	266
13.4.1 fontinst 巨集套件	266
13.4.2 選用字體	271
13.4.3 使用 True Type 字體	272
13.5 DVIPS 與 psutils 工具程式	272
13.5.1 DVIPS 程式	272
13.5.2 afm2tfm 程式	273
13.5.3 psutils 工具程式	274
14 巨集指令	277
14.1 定義巨集指令	277
14.1.1 設定字級之巨集指令	279
14.1.2 巨集指令與中文	281
14.1.3 依條件處理之巨集指令	281
14.2 定義指令環境	282
14.3 更改內定之標題為中文	283
14.4 計數器	283
15 有用的工具套件	287
15.1 信函	287
15.1.1 排版信函指令	287
15.1.2 設計個人信頭標誌	289
15.1.3 中文信頭標誌	291

15.2	大宗信函	295
15.3	固定格式標籤	297
15.4	投影片	299
15.4.1	FoilTeX 巨集套件	300
15.4.2	數學式	301
15.4.3	圖形與彩色	302
15.5	Power Point 樣式投影片	302
15.5.1	pdfscreen 巨集套件	303
15.5.2	特殊效果	305
15.5.3	引用 PDF 格式圖形檔	306
15.6	習題與解答	307
16	參考文獻	311
16.1	排版參考文獻之流程	311
16.2	調整設定	314
16.3	建立文獻資料檔	315
16.4	natbib 巨集套件	318
16.4.1	徵引文獻之指令	318
16.4.2	流通文稿	320
16.5	製作文獻格式檔	321
16.6	listbib 巨集套件	321
17	版面設計	323
17.1	geometry 巨集套件	323
17.2	titlesec 巨集套件進階指令	326
17.2.1	彩色或灰階色標題文字	329
17.2.2	章節標題排版例子	330
17.2.3	嵌入標題	333
17.2.4	序文與索引之標題	335
17.3	titletoc 巨集套件	335
17.4	fancyhdr 巨集套件	338
17.4.1	單雙頁相同版面	339

17.4.2	單雙頁版面不同	341
17.4.3	頁眉與頁足之章節標題	342
18	書籍排版	347
18.1	排版設計	347
18.2	書籍排版例子	349
18.3	索引	353
18.3.1	標註索引名詞	353
18.3.2	排版索引的步驟	355
18.4	裁切記號	358
19	網路出版	359
19.1	HTML 與 PDF 的比較	359
19.2	轉換為 PDF 格式	360
19.2.1	Ghostscript	361
19.2.2	pdf \LaTeX	361
19.2.3	Acrobat Distiller	362
19.2.4	hyperref 巨集套件	362
19.3	\LaTeX 2HTML	364
19.3.1	安裝	364
19.3.2	使用方法	367
19.3.3	轉換中文稿	369
20	造字	373
20.1	轉換 TrueType 字型檔為 $\text{cw}\LaTeX$ 字型檔	373
20.1.1	cwpt1 與 cwttf 軟體	373
20.1.2	轉換方法	374
20.1.3	字型對應檔	376
20.1.4	選用字體名稱	378
20.2	中文字之排序	379
20.3	造新字	380
20.4	使用新字	380

21 錯誤訊息	383
21.1 cwtext 訊息	383
21.2 latex 訊息	385
21.3 確認錯誤來源	387
22 資源	389
參考書目	391
數學符號索引	393
索引	399

修訂版重要更新

本書為 cwTeX 2 版修訂。為行文方便，底下將以 2.1 版稱呼。2.1 版與第 2 版的主要差異如下：

- 中文字體指令使用 TeX 之 New Font Selection Scheme (NFSS)，
- 提供 cwbibtex 軟體以排版中英文參考文獻，
- 介紹如何製作類似 Power Point 之投影片檔案，
- $\backslash\text{verb}$ 指令與 verbatim 指令環境可處理中文，
- 提供字型工具程式，可將 Windows 之中文字型轉換為 cwTeX 字型，
- Linux 版本全新改寫，與 Windows 版本具相同功能。

最後兩項是翁鴻翎先生的貢獻。對一般使用者而言，第一項的改變對於中文排版可能最為重要，以下略作說明。

在第 2 版中，若文稿以 $\backslash\text{title}$ 指令排版中英文標題， TeX 會自動將英文字適當放大，但中文字則維持與正文字體相同。因此，排版者須另外以中文字體指令改變中文字之字體與大小。譬如，若選用粗黑體 20 級字體，須加入指令 $\backslash\text{bb}20$ 。新版 cwTeX 中， $\backslash\text{title}$ 指令除了改變標題之英文字外，同時也將中文字調整到與英文字相同之大小；不過，中文字體仍選用內定之明體字。因此，若標題並不選用不同字體，但希望其大小與英文字匹配，我們不須加上任何指令。如果標題內要選用特定之中文字體，如粗黑體，指令為 $\backslash\text{ctxfbb}$ 。因此，在新版 cwTeX 之下，排版者僅須選用字體，大小則交由 TeX 控制調整。

另外一個例子是排版註腳。 TeX 之 $\backslash\text{footnote}$ 指令會自動選用較小之英文字體。但在舊版中，中文字體不會隨之而變小，故必須自行加上指令，如 $\backslash\text{m}10$ ，以控制中文字體之大小。在新版系統下， $\backslash\text{footnote}$ 指令同時使中英文字縮小，我們不須再另外以中文字體指令調整。安裝完成之後，請由 $\text{c}:\backslash\text{texmf}\backslash\text{cwtex}\backslash\text{examples}$ 檔案夾內各個例子了解新版中文字體指令使用法。

Linux 版本是由翁鴻翎先生完成, 李果正先生幫忙測試, 並提供意見。我們特別感謝他們的熱心協助。此外, 排版參考文獻之 `cwibibtex` 程式, 及中文 True Type 字型轉換成 `cwTeX` 字型檔之工具程式也是翁鴻翎先生的傑作。本書完稿前夕, 沈俊興先生幫忙 compile FreeBSD 平台之版本。此一版本尙有待進一步測試, 等到穩定之後, 將置於網站上供下載。

`cwTeX` 出版之後, 許多熱心的使用者提供改進意見, 我們非常感激。在時間、能力許可的情況下, 我們儘可能把讀者的意見納入新版內。不過, 我們的時間與精力有限, 一些寶貴意見只能留待未來。其中, Mobydick 所提供的 `cwTeX` 字型筆劃之修改意見, 只能等待下一個修正版了。本書準備出版時, 楊孝霖小姐仔細閱讀本書前8章, 挑出許多錯誤, 我們在此表示感謝。

吳聰敏·吳聰慧

2002.4

2 版序

cwTeX 系統從開始發展迄今，已超過十年。開始發展此一系統的動機很簡單：因為在學術界的工作需要一套排版學術論文的軟體。幾年之後，我們才發現 Knuth 教授當初也是因為同樣的理由而創造 TeX 系統。

cwTeX 系統是以 TeX 為骨幹。發展軟體時，我們希望它的能力、品質與英文 TeX 系統相當。但是，中文書寫系統與英文畢竟差異很大，在某些地方我們實在是找不出好的解決之道。不過，我們認為 cwTeX 的能力與排版品質，相較於目前任何中文排版系統都毫不遜色。這當然是要歸功於 TeX 幾近完美的設計。

本書初版於 1997 年出版以來， \LaTeX 排版系統有許多新的發展，其中最重要的可能是網路出版；因此，以 \LaTeX 作為網路出版的工具程式也陸續出現。利用這些軟體工具， \LaTeX 的排版結果很容易可以送上網站。因應此一發展，本版對於網路出版另闢一章介紹。除了網路出版之外，第 2 版在許多地方與初版差異甚大，簡單說明於下：

- cwTeX 中文字體指令之語法改為與 \LaTeX 一致。譬如，中文字體指令有效範圍僅限於指令環境或大括號內；另外，中文字體指令之後可續鍵入文字，不須另換新行。
- cwTeX 提供簡單的中文字體巨集指令功能，用於設定文稿章節標題與註解之中文字體。
- 每一中文字體可以作斜體與水平放大/縮小之變形。
- 新版 DVIPS 能正確處理中文 PostScript 字型檔，因此文稿排版結果可直接以描邊字型列印。
- 介紹一些新的巨集套件，使文稿編排設計更為容易。譬如，`titlesec` 巨集套件使章節標題設計變得容易，且富有彈性。又如，`fancyvrb` 巨集套件之 `Verbatim` 指令環境可以照列中文與排版指令。

本書初版發行時，Win95 作業系統雖已出現，但專以此作業系統為對象之 T_EX 系統尚不十分成熟。兩年來，專門在 Win95/98 系統上使用之 T_EX 系統包括 MiK_TE_X 與 f_pT_EX 兩套。目前，c_wT_EX 可以在這兩套系統上使用。除了 Win95/98 之外，c_wT_EX 也可以在 Windows NT 或 Windows 2000 上執行。事實上，除了 Windows 平台之外，一些熱心人士正幫忙將 c_wT_EX 移植到 Linux, FreeBSD 平台上。如果有新的進展，我們將在 c_wT_EX 網站上公佈。

使用 c_wT_EX 給我們帶來相當的便利，甚至是樂趣；我們希望你也有同樣的感受。發展軟體的過程中，張清溪、林向愷、鍾經樊、古慧雯與宋玉生教授提供許多寶貴意見。歷年來台大經研所的研究生不少人以此系統排版論文，等於是幫忙測試軟體。其中，盧敬植、曲祉寧、陳宜廷等特別熱心幫忙。台大經濟系、淡大數學系及嘉南藥理學院提供硬體設備與軟體；陳俞成先生細心校閱初稿；吳慕凡小姐繪製插圖；我們很感激他們的幫助。台大經濟系的研究助理陶曉昀、陳香如、徐中貴、楊文琦、湯雅慧等，特別是張素惠小姐，對於發展此一系統貢獻良多，我們在此一併致謝。

準備第2版過程中得到許多人的協助。首先，許多的熱心使用者提出各種建議，但限於時間與能力，部分建議無法完成，謹在此表示感激與歉意。朱雅珍從美國寄來參考書籍，陳俞成再度細心校閱初稿，翁明德積極推薦 f_pT_EX，並提供實質幫助；他們的協助使得軟體與手冊的品質得以大幅提升。最後，沒有林佳蓉與曲祉寧的辛苦工作，本書的改版是難以完成的。我們也藉此機會特別表示感謝。

幾十年來，父母樸素、平實的生活態度一直是我們的榜樣。他們給予子女們的持續關懷，對於我們的工作，包括發展 c_wT_EX 系統在內，有積極的鼓勵作用。我們也在此表達感謝之意。

本版初稿完成之際恰逢台灣遭受百年來最大的集集大地震（1999年9月21日凌晨1時47分）。謹在此誌念地震中不幸傷亡的人們。

吳聰敏·吳聰慧

1 前言

本書介紹 cwTeX 排版系統。 cwTeX 包含 TeX 排版系統，中文字轉換程式與各式中文字體。 TeX 系統是美國 Stanford 大學 Donald E. Knuth (高德納) 教授所發展，目前是美國數學學會 (American Mathematical Society) 出版學術期刊與書籍的排版系統。一般認為 TeX 的特點是排版數學文稿能力特別強。不過，根據 Knuth 的夫子自道：¹

“[T]he TeX research project that I embarked on was driven by two major goals. The first goal was *quality*: we wanted to produce documents that were not just nice, but actually the best.”

“I never intended to have a system that would be universal and used by everybody. I always wanted to write a system that would be used for just the finest books.”

顯然，品質才是 TeX 系統的真正精神。

cwTeX 以 TeX 為基礎，但加強其功能，使它可以排版中文。本書說明如何使用 cwTeX 排版中英文書籍與文稿。 TeX 的排版指令多而且複雜。對於專業排版者來說，這有「如魚得水」的感覺。但對於一般的使用者，可能望之而卻步。為了因應一般使用者的需求，許許多多的專家寫了一套套的巨集指令 (macros)。利用這些現成的巨集指令，以 TeX 排版書籍稿件並不困難。各巨集指令中，最有名也最廣為使用的是 \LaTeX ，原始作者為 Leslie Lamport。本書主要介紹 \LaTeX 指令。

Knuth 教授原先是在大型電腦上發展 TeX 系統。完成之後，他把原始程式全部公開。因此有人把 TeX 移植 (porting) 到各種作業系統上。目前，大至超級電腦，小至個人電腦上，都有 TeX 系統。其中，大部分都是免費使用。本書初版出版於 1997 年，當時雖然許多個人電腦所使用的作業系統已逐漸由 MSDOS 與 Win31 轉為 Win95，但是專屬於 Win95 之 TeX 系統尚未成熟。因此，本書初版所介紹的是使用於 MSDOS 作業系統上之 emTeX 。幾年來，專以 Windows 系統

¹見 Knuth (1999)，頁559, 616。

為對象的 $\text{T}_\text{E}\text{X}$ 已有好幾套，其中可供免費下載的包括 $\text{MiK}\text{T}_\text{E}\text{X}$ 與 $\text{fp}\text{T}_\text{E}\text{X}$ 兩套。目前， $\text{cw}\text{T}_\text{E}\text{X}$ 主要即結合這兩套系統發展。

以上所介紹的與 $\text{T}_\text{E}\text{X}$ 有關的軟體可簡單歸納如下：

- $\text{T}_\text{E}\text{X}$: Knuth 所發展之排版系統；
- $\text{E}\text{T}_\text{E}\text{X}$: 方便使用之巨集套件組合；
- $\text{MiK}\text{T}_\text{E}\text{X}/\text{fp}\text{T}_\text{E}\text{X}$: 可於 Windows 平台上執行之 $\text{T}_\text{E}\text{X}$ 系統；
- $\text{cw}\text{T}_\text{E}\text{X}$: 讓 $\text{T}_\text{E}\text{X}$ 可以排版中文稿件之工具程式與字型檔案。

要讓 $\text{T}_\text{E}\text{X}$ 處理中文，基本上有兩種方法。第一是直接修改 $\text{T}_\text{E}\text{X}$ 原始程式碼。我們採取的是較間接的第二種方法：寫一個中文字碼轉換程式（亦稱為前階處理程式 *preprocessor*）， cwtex 。此程式的主要功能是將中文字碼轉換成 $\text{T}_\text{E}\text{X}$ 的格式。我們另外造出符合 $\text{T}_\text{E}\text{X}$ 規格的各式中文字型。如此一來， $\text{T}_\text{E}\text{X}$ 即可以編排中文字。此種方法較簡單，而且程式出現錯誤的機率較低。

我們採取前階處理程式來處理中文還有另外一個原因。完整的排版過程包括輸入文字及指令、編排（*text formatting*）、預視與列印。要在顯示器上預視排版結果需要預覽軟體；在不同印表機上列印也需要特別的列印程式。自從 $\text{T}_\text{E}\text{X}$ 出現之後，很多人寫了預覽/列印軟體及相關的工具程式（*utilities*），絕大部分是免費提供使用。 $\text{cw}\text{T}_\text{E}\text{X}$ 前階處理程式轉換中文時，完全依照 $\text{T}_\text{E}\text{X}$ 之規格。因此，幾乎所有的預覽/列印軟體及工具程式都可用於中文稿件上，不須修改。

$\text{E}\text{T}_\text{E}\text{X}$ 是由一大群巨集指令組合而成。從開始發展以來，許多使用者寫了更多的巨集指令擴充其功能。目前有一群專家正在更新 $\text{E}\text{T}_\text{E}\text{X}$ 巨集指令。此一工作完成之後，將稱為 $\text{E}\text{T}_\text{E}\text{X}$ 第3版，目前流通之版本稱為 $\text{E}\text{T}_\text{E}\text{X}2\epsilon$ ；舊有版本則稱為 $\text{E}\text{T}_\text{E}\text{X} 2.09$ 。本書所介紹的是 $\text{E}\text{T}_\text{E}\text{X}2\epsilon$ 之指令。

除了 $\text{E}\text{T}_\text{E}\text{X}$ 及 $\text{cw}\text{T}_\text{E}\text{X}$ 中文指令外，本書也扼要介紹一些排版的觀念。文稿排版品質的好壞，除了排版系統的能力之外，更重要的是排版者的鑑賞能力。 $\text{E}\text{T}_\text{E}\text{X}$ 的排版能力一流。但是，頂級的咖啡機器（如 *Rancilio Audrey*）若使用不當，也可能燒出如 *Agatha Christie* 所說的「名為咖啡的可疑液體」。我們從排版文獻中整理出一些應注意的事項。了解這些基本觀念，排版時可以避免一些常見的錯誤。

本書是 $\text{cw}\text{T}_\text{E}\text{X}$ 排版系統之使用手冊。學習電腦排版，必須實際動手，不能光說不練。要學習使用 $\text{cw}\text{T}_\text{E}\text{X}$ 系統，請先安裝程式。第4章說明如何安裝 $\text{cw}\text{T}_\text{E}\text{X}$ 系

統。初學習排版，請先閱讀 2–6 章。第 2 章提供 7 個排版例子，第 3 章說明完整的排版過程；你可以依樣輸入文稿、排版、預視與列印。第 4 章除了說明安裝軟體的方法之外，也介紹如何透過 WinEdt 軟體排版。第 5 章說明排版指令的概念及輸入文稿的原則。第 6 章說明選用中英文字體的方法。

第 7、8 兩章說明章節與段落設計指令。若是排版書籍，我們首先須設計好版面樣式，再以這兩章的指令編排。E_TX 提供一些現成的版面格式，可供直接使用。第 9 章說明數學式之排版，對某些人來說，這可能是 T_EX 最引人入勝之處。第 10 章為表格編排。表格排版指令較複雜，因此我們提供較多的例子說明。

新版 E_TX 一個重要的發展方向是與 PostScript 繪圖語言結合。此一結合讓我們可以使用各種專業品質之英數字體、引用外製圖形、或者在文稿中直接繪製圖形。第 11–13 章即介紹相關之指令與巨集套件。

E_TX 事實上是以前 T_EX 為骨幹的一套巨集指令，第 14 章說明巨集指令之概念及使用方法。第 15 章介紹一些有用的工具，包括：排版信函與排版投影片等等。一般的論文寫作不可避免地須引用參考文獻。E_TX 提供一套排版參考文獻的工具程式，稱為 bibtex。中文參考文獻的格式與英文不同，故 cw_ET_EX 特別提供 cw-bibtex 程式以便能同時排版中英文參考文獻。第 16 章即介紹此一系統。

T_EX 系統的能力足以排版任何長度的複雜書籍。書籍排版時，我們須考慮種種的版面設計問題。第 17 章介紹版面設計時常使用的幾個巨集套件。18 章則以實際例子說明如何排版書籍，我們同時也介紹排版索引的方法。第 19 章說明網路出版的相關工具與軟體。cw_ET_EX 系統所提供的字型檔大部分都含有 13,503 個中文字。即使如此，我們偶而還是會碰到一些字型檔內所沒有的中文字。遇有此種狀況，必須自行造字。第 20 章說明造新字的方法。

cw_ET_EX 為幕後排版系統，下指令或輸入文字時，難免出現錯誤。第 21 章說明如何偵測錯誤。最後，第 22 章說明如何下載並使用新出版之巨集套件。

* * *

本書的排版幾乎使用了本書所介紹的各種巨集套件與程式。一開始，我們是分章編排。到了最後階段，我們創造一主檔案，取名為 cxbook.ctx，其內引入全書各章。排版過程中，我們同時也測試 MiK_ET_EX 系統之功能。因為全書內容複雜，執行 latex 時，我們必須把 font_mem_size 由原設定值 200000，加大為 300000，

否則會出現錯誤。另外, 可能是因為使用字型檔太多的原因, DVIPS 無法將 `cxbook.dvi` 轉換為 `cxbook.ps`。我們是藉助 Omega 系統之 `odvips`, 才能成功轉換出 PostScript 檔。

`latex` 執行後會產生 `cxbook.log`, 其中包含執行過程中的種種訊息。但是, MiKTeX 系統下之 `latex` 所產生的訊息較少。以下是以 `emTeX` 之 `htex386` 重跑之後所產生之訊息:

```
Here is how much of TeX's memory you used:
6984 strings out of 55921
73070 string characters out of 211291
144779 words of memory out of 2097153
8556 multiletter control sequences out of 28000
281635 words of font info for 532 fonts,
  out of 400000 for 759
14 hyphenation exceptions out of 2437
33i,23n,46p,2711b,688s stack positions
  out of 1000i,100n,60p,50000b,1000s
```

2 例子

學習排版的捷徑是參考現成的例子。本章列出 7 個排版例子，並加上簡單的說明。這幾個例子之檔案皆置於 `c:\texmf\cwtex\examples` 檔案夾內。若自行排版這幾個例子，得到的結果與本章所展示者會稍有差異，原因是本書排版使用了一套特別的英文字體。

除了以上 7 個例子外，檔案夾內尚含有其他例子可供參考。其中，排版一般專業論文可參考 `paper1.ctx` 與 `paper2.ctx`；`thesis1.ctx` 與 `thesis2.ctx` 是研究生論文的例子。E_T_X 系統可用以排版信函，`letter1.ctx`，`...`，`letter3.ctx` 是排版信函之例子，請見 15 章之說明。`ntueltr.ctx` 與 `ntueltr.cls` 說明如何製作信頭標誌。`foil-1.ctx` 與 `foil-pp4.ctx` 是排版投影片的例子。若有意自行排版書籍，請參考 `c:\texmf\cwtex\examples\book` 子檔案夾內例子，主檔案為 `ecobook.ctx`。

此外，`NSCproj.ctx` 是國科會研究補助之申請表格；`NSCvita.ctx` 是個人資料表。國科會表格經常變動，以上檔案所排版的格式與國科會最新格式可能稍有差異。最後，`test-h1.tex`，`test-h2.ctx`，`...`，與 `test-h5.ctx` 是由 E_T_X 檔案轉換為 HTML 檔案的例子，排版方法請見 19 章之說明。

國民所得兩萬美元

吳聰敏

2002.1

台灣的生活水準在提升嗎？若從大街小巷充斥著賣名牌的商店來看，答案是肯定的。但如果從生活環境品質日益惡化來看，答案則是否定的。

三十年前，台灣的所得尚低。一般民衆最關心如何提升所得；因此政府施政也以提升所得為主要目標，這不難理解。但是，所得逐漸上升之後，民衆的偏好、需求會逐漸改變。每一個國家都是如此，台灣也不例外。最簡單的一個例子，幾年來出國旅遊風氣日盛，這表示民衆對休閒的需求日增。因為台灣本島可供旅遊休閒的場所太少，大家只好往國外跑。（但是，並不是每一個人都有能力到國外渡假。）

公元兩千年時，台灣每人平均國民所得高達兩萬美元。上下班時，每人身著名牌，在公車或自用車子裡動彈不得，空氣品質及水質比今天更惡化；遇到「假日」除非有錢出國，否則只好呆在客廳裡吹冷氣、看電視。這樣子的兩萬美元所得的意義在那裡呢？


```
% file: exampl.ctx
\documentclass[12pt]{article}
\title{\ctxfbb 國民所得兩萬美元}
\author{\ctxfk 吳聰敏}
\date{2002.1}
\begin{document}
\maketitle
\fontsize{12}{20pt}\selectfont
台灣的生活水準在提升嗎？
若從大街小巷充斥著賣名牌的商店來看，
答案是肯定的。
但如果從生活環境品質日益惡化來看，
答案則是否定的。

三十年前，台灣的所得尚低。
一般民衆最關心如何提升所得；
因此政府施政也以提升所得為主要目標，
這不難理解。
但是，所得逐漸上升之後，民衆的偏好、需求會逐漸改變。
每一個國家都是如此，台灣也不例外。
最簡單的一個例子，幾年來出國旅遊風氣日盛，
這表示民衆對休閒的需求日增。
因為台灣本島可供旅遊休閒的場所太少，
大家只好往國外跑。
(但是，並不是每一個人都有能力到國外渡假。)

公元兩千年時，台灣每人平均國民所得高達兩萬美元。
上下班時，每人身著名牌，
在公車或自用車子裡動彈不得，
空氣品質及水質比今天更惡化；
遇到「假日」除非有錢出國，
否則只好呆在客廳裡吹冷氣、看電視。
這樣子的兩萬美元所得的意義在那裡呢？
\end{document}
```

- 排版之前須先決定文稿結構。L^AT_EX 以 `\documentclass` 選用文件類別。本例選用 `article` 文件類別。文件類別一旦決定，文稿結構也即確定。故文稿第一道指令通常是 `\documentclass`，我們稱之為文件類別指令。指令內加上 `[12pt]` 指示以 12pt 字體排版。文稿正文內容置於 `document` 指令環境內，以 `\begin{document}` 起頭，以 `\end{document}` 結束。
- 排版指令皆以反斜線 `\` 起頭，但註銷指令是 `%`。
- 題目排版於文稿前端，但不屬於正文範圍。`\title` 指令用於排版標題，`\author` 用於排版作者名字，`\date` 用於排版日期。輸入排版題目之指令與文字之後須加上 `\maketitle` 指令，指示將以上之文字全部排版於標題頁或標題區域內。
- `\ctxfbb` 與 `\ctxfk` 為中文字體指令，分別表示選用粗黑字體與楷體中文。
- 文件類別指令之 `[12pt]` 選項在選定字體大小時，同時也決定了行距。對中文稿而言，L^AT_EX 內定之行距太小，故我們再以 `\fontsize` 指令選定行距為 20pt。
- 空一行表示本段結束。輸入文稿時，請盡量在標點符號後換行。

台灣長期總產出之變動

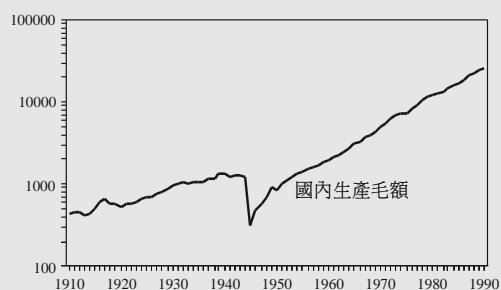
吳聰敏*

2001.3

台灣長期總體經濟表現，可以簡單分清國統治時期、日治時期及戰後時期來討論。其中，清末日初是一個重要的轉捩點。

1 長期國內生產毛額之變動

百年來台灣的國民所得有很大的變化。下圖畫出台灣長期國民所得之變動。¹由此圖可以看出，日治時期國民所得成長率低於戰後時期。



2 資本主義化

日治初期總督府推動**資本主義化**的政策，其目的是要改善台灣的投資環境，吸引日本企業家前來投資。日本人在台灣進行土地與戶口調查，開闢道路，統一貨幣與度量衡。吸引日本企業家前來投資。

*作者任教於台大經濟系。

¹日本人曾粗略估算台灣1938-44年之國民所得，刊於《台灣金融經濟月報》1944年8月號。

```

% file: examp2.ctx
\documentclass[11pt]{article}
\usepackage{graphicx,psfrag}
\linespread{1.2}

\title{ 台灣長期總產出之變動}
\author{ 吳聰敏\thanks{
作者任教於台大經濟系。}}
\date{\small 2001.3}
\begin{document}
\maketitle
\noindent
台灣長期總體經濟表現，
可以簡單分清國統治時期、
日治時期及戰後時期來討論。
其中， 清末民初是一個重要的轉捩點。

\section{ 長期國內生產毛額之變動}
百年來台灣的國民所得有很大的變化。
下圖畫出台灣長期國民所得之變動。
\footnote{
日本人曾粗略估算台灣1938-44 年之國民所得，
刊於《台灣金融經濟月報》1944 年8月號。}
由此圖可以看出，
日治時期國民所得成長率低於戰後時期。

\begin{figure}[h]
\centering
\psfrag{GDP}{\small 國內生產毛額}
\includegraphics[width=.7\textwidth]{examp.pdf}
\end{figure}

\section{ 資本主義化}
日治初期總督府推動{\ctxfbb 資本主義化}的政策，
其目的是要改善台灣的投資環境，
吸引日本企業家前來投資。
日本人在台灣進行土地與戶口調查， 開闢道路，
統一貨幣與度量衡。
吸引日本企業家前來投資。
\end{document}

```

- 文件類別指令之 [11pt] 選項指示選用 11pt 字體。
- `\usepackage` 指令用於引入巨集套件。本例子引入 `graphicx` 與 `psfrag` 兩個巨集套件，其功能是引用圖形。在 `\begin{document}` 之前的區域稱為全文設定區 (preamble)。引入巨集套件之指令須下於全文設定區。
- 實際引入圖形之指令是 `\includegraphics`，指令中又加入 `width=` 選項，指示將圖形放大或縮小於行寬的 70%。原圖中有 “GDP” 一字，`\psfrag` 指令將之替換為「國內生產毛額」。
- \LaTeX 內定之行距適用於英文稿，但不適用於中文稿。中文字筆劃較複雜，行距須加大。`\linespread{1.2}` 指令將行距加大為內設值的 1.2 倍。
- `\thanks` 用於排版致謝詞。
- `\date` 指令內加入 `\small` 指令，指示選用較小之英數字體。
- `\section` 指令用於排版節標題，標題之文字會自動放大一些，前後並多留出一點空白。
- 排版註解指令為 `\footnote`， \LaTeX 會自動替章節標題與註解加上數字編號，並選用較小的字體。

戰後初期，國民政府在台灣實施全面性的經濟管制，其手段包括：民營企業公營化、獨占與聯合壟斷、價格控制等。

1 文獻檢討

有關於台灣戰後初期經濟的文獻很少，連溝口敏行的專著，《台灣・朝鮮經濟の成長》，也未討論此一時期的經濟狀況。日本大藏省所出版的《日本人の海外活動に關する歷史的調查》，則收集不少此時期的統計資料。

2 貿易管制

1946–48年間，台灣的砂糖絕大部分銷往上海。國民政府極力壓低砂糖的銷售價格。價格一受到管制，台糖的銷售收入也大受影響。表1列出台灣出口至大陸的糖、煤及總出口之物價指數。

表 1: 台灣對大陸出口金額

	1946	1947	1948
糖	100	392	436
煤	100	455	4,736
物價指數	100	463	2,871

說明：金額單位為台幣百萬元。

根據台糖公司本身的估計，干預政策使台糖在1947–49年間，「損失達新臺幣一億四千餘萬元」。事實上，國民政府的管制政策是戰後初期台灣惡性物價膨脹的主要因素之一。

examp3.ctx (January 17, 2002)

```

% file: examp3.ctx
\documentclass[12pt]{article}
\usepackage{booktabs}
\input{mymacro}
\renewcommand{\tablename}{ 表}
\ctxfdef{\section}{\ctxfr}

\begin{document}
\sz12
戰後初期，國民政府在台灣實施全面性的經濟管制，...

\section{ 文獻檢討}
有關於台灣戰後初期經濟的文獻很少，
連溝口敏行的專著，《台灣  $\cdot$  朝鮮經濟 成長》，
...

\section{ 貿易管制}
1946—48 年間，...
表\ref{export}\Z 列出台灣出口至大陸的糖、
煤及總出口之物價指數。

\begin{table}[h]
\begin{center}
\caption{ 台灣對大陸出口金額} \label{export}
\begin{tabular}{rrrr}
\toprule
& 1946 & 1947 & 1948\\
\midrule
糖 & 100 & 392 & 436\\
煤 & 100 & 455 & 4,736\\
物價指數 & 100 & 463 & 2,871\\
\bottomrule
\end{tabular}
\par\smallskip
\parbox{.45\textwidth}{ 說明：金額單位為台幣百萬元。}
\end{center}
\end{table}

根據台糖公司本身的估計，...

\par\vfill\noindent \jobname.ctx (\today)
\end{document}

```

- 排版表格可以使用 \LaTeX 之 `tabular` 指令環境。全文設定區內引入 `booktabs` 巨集套件，其指令使表格排版更精美。
- 全文設定區內又以 `\input` 指令引入 `mymacro.tex`，其內定義選擇行距之指令。利用其定義，`\sz12` 指令可選用 12pt 字體及行距。
- 排版表格指令置於 `table` 指令環境內， \LaTeX 會在版面中找出置放表格最適當之處。指令環境內可使用 `\caption` 排版標題。 \LaTeX 自動替表格編號，若是第 1 表，標題前端會加上“Table 1”。我們在全文設定區使用 `\renewcommand` 指令將“Table”改為“表”。
- 欲引述某一表格我們可使用 `\label` 與 `\ref` 指令。本例中，`\caption` 指令之後加上 `\label{export}`，排版時， \LaTeX 自動算出 `export` 之數值為 1。文稿他處以 `\ref{export}` 指令即可正確引述表 1。`\ref` 指令前後加入 `\Z` 指令，目的是使中文與阿拉伯數字之間距稍微加大一點。
- `cwTeX` 可以排版日文假名。
- 全文設定區之 `\ctxfdef` 是 `cwTeX` 中文字體指令，可選定節標題 `\section` 內之中文字體。本例選用 `\ctxfr` 圓體字。
- 文稿末端之 `\jobname` 指令用於排版文稿檔名，`\today` 用於排版日期。

論「米糖相剋」

古慧雯·吳聰敏*

2001.3

稻米和蔗糖是日治時期最重要的產業，文獻上所謂的「米糖相剋」問題，是在探討甘蔗和稻作競地的情形。

1 模型

影響作物選擇的因素很多，包括各作物的單位價格及其生產力。根據以上的分析，農夫選種甘蔗的條件是：

$$\frac{q_C^k(t)}{q_{R1}^k(t)} \geq \frac{P_{R1}(t)}{P_C(t+3)} \frac{1 + \delta + \delta^2}{\delta^2} \quad (1)$$

很明顯地， t 時種蔗的總人口或總面積，繫乎其時各農夫蔗稻的相對生產力，以 $F_t(\cdot; \alpha_t)$ 表示 $q_C^k(t)/q_{R1}^k(t)$ 之累積分配函數，其中參數 α_t 反映農業政策對蔗稻相對生產力的衝擊。 α_t 值較大時，蔗稻相對生產力的分配亦較具「隨機優勢」，亦即新政策有利於蔗作勝過稻作： $\partial F_t / \partial \alpha_t < 0$ 。

由式 (1)， t 時甘蔗植付面積之比率為：

$$1 - F_t \left(\frac{P_{R1}(t)}{P_C(t+3)} \frac{1 + \delta + \delta^2}{\delta}; \alpha_t \right)$$

*感謝楊文琦、徐中貴、賴香吟小姐協助搜集資料。

```

% file: examp4.ctx
\documentclass[11pt]{article}
\input{mymacro}
\ctfdef{\section}{\ctxff}
\usepackage{txfonts}

\title{\ctxff 論「米糖相剋」}
\author{古慧雯。吳聰敏 \thanks{
感謝楊文琦、徐中貴、賴香吟小姐協助搜集資料。}}
\date{\small 2001.3}
\begin{document}
\maketitle
\sz11
稻米和蔗糖是日治時期最重要的產業，
文獻上所謂的「米糖相剋」問題，
是在探討甘蔗和稻作競地的情形。

\section{模型}
影響作物選擇的因素很多，
包括各作物的單位價格及其生產力。
根據以上的分析，農夫選種甘蔗的條件是：
\begin{equation}
\frac{q_{C^k}(t)}{q_{R1}^k(t)} \geq
\frac{P_{R1}(t)}{P_C(t+3)}
\frac{1+\delta+\delta^2}{\delta^2}
\end{equation}
很明顯地， $t$  時種蔗的總人口或總面積，
繫乎其時各農夫蔗稻的相對生產力，
以  $F_t \cdot \alpha_t$  表示
 $\frac{q_{C^k}(t)}{q_{R1}^k(t)}$  之累積分配函數，
...
亦即新政策有利於蔗作勝過稻作：
 $\frac{\partial F_t}{\partial \alpha_t} < 0$  。

由式 (1)， $t$  時甘蔗植付面積之比率為：
\[
1 - F_t \left( \frac{P_{R1}(t)}{P_C(t+3)}
\frac{1+\delta+\delta^2}{\delta} ;
\alpha_t \right)
\]
\end{document}

```

- T_EX 提供相當豐富的數式字體，稱為 Computer Modern 字體。本例特別引用 txfonts 巨集套件，這是 Young Ryu 所設計的另一整套數式字體。其字體風格接近 Times Roman。
- 排版數學式之指令須置於數學模式中，本例分別使用 equation 指令環境與 $[\dots]$ 編排展示數式；前者有自動編號之功能。隨文數式或數學符號則以 $\$ \dots \$$ 排版。
- \title 指令內加入 \ctxff 以選用仿宋體。 \date 指令內加入 \small 選用較小字體。
- 數學符號是以指令排版，譬如 α_t 產生 α_t 。又如， δ^2 排版為 δ^2 。

從生產面計算國內生產毛額時，通常把產業區分為：(1) 農畜業、(2) 製造業與 (3) 服務業。

1 統計資料之性質與來源

農畜業主要產品生產額之原始資料來源主要為《總督府統計書》與《臺灣農業年報》。

2 農畜業

農畜業分普通作物、特用作物等五大類。

2.1 普通作物

普通作物含米穀、甘藷等項。蓬萊米自 1920 年中期開發成功之後，其產量在短時間內就凌駕在來米之上。但是，1946-48 之間在來米產量反而高於蓬萊米。戰後初期，台灣化學肥料極為缺乏；影響所及農民只好暫時改種在來米。¹

2.2 特種作物

特種作物主要為甘蔗與粗製茶。1902 年之甘蔗生產額仍低於粗製茶；但翌年開始，甘蔗生產額即超過粗製茶，1910 年代開始，糖業之繁榮使特種作物生產額比率大幅上升，最高約達 80%。

¹請見于景讓 (1949)，頁 14-16。


```
% file: examp5.ctx
\documentclass[11pt]{article}
\ctxfdef{\section}{\ctxfbb}
\ctxfdef{\subsection}{\ctxfbb}

\begin{document}
\fontsize{10.95}{18pt}\selectfont
從生產面計算國內生產毛額時，
通常把產業區分為：(1) 農畜業、(2) 製造業與 (3) 服務業。

\section{ 統計資料之性質與來源}
農畜業主要產品生產額之原始資料來源主要為
《總督府統計書》與《臺灣農業年報》。

\section{ 農畜業}
農畜業分普通作物、特用作物等五大類。

\subsection{ 普通作物}
普通作物含米穀、甘藷等項。
蓬萊米自1920 年中期開發成功之後，
其產量在短時間內就凌駕在來米之上。
但是，1946-48 之間在來米產量反而高於蓬萊米。
戰後初期，台灣化學肥料極為缺乏；
影響所及農民只好暫時改種在來米。
\footnote{
請見于景讓 (1949)，頁14--16 。}

\subsection{ 特種作物}
特種作物主要為甘蔗與粗製茶。
1902 年之甘蔗生產額仍低於粗製茶；
但翌年開始，甘蔗生產額即超過粗製茶，
1910 年代開始，
糖業之繁榮使特種作物生產額比率大幅上升，
最高約達80%。
\end{document}
```

- 文稿如果不排版題目，則直接以 `\begin{document}` 開始排版正文。
- 短文之章節結構分：`\section`，`\subsection`，`\subsubsection`，`\paragraph`，與`\subparagraph`。如果是使用 `book` 文件類別，我們還可使用 `\chapter` 排版每章之標題。本例僅使用節與小節標題指令。
- `cwTeX` 的中文字體巨集指令 `\ctxfdef` 可用來選用章節標題之中文字體。如果註解要以仿宋體排版，只要在全文設定區以 `\ctxfdef` 指令定義即可。
- 本例設定節與小節標題都使用粗黑體。排版時，節標題會放大一些；小節標題之字體也比正文大；但比節標題小一些。

台灣農畜業之生產額: 1902–52

吳聰敏

2002.3.1

根據《中華民國國民所得編算方法說明》(1969, 頁18–22), 農畜業生產額由下列三大項加總而得: (1) 農耕生產總額, (2) 畜禽生產總額, (3) 其他收入。依 *A System of National Accounts* (1993, 頁150–52), 產品生產額應以 basic price 乘上數量求算。本研究農畜業產品生產額資料主要取自《臺灣省五十一年來統計提要》(1946)。

日本投降之後, 國民政府接收所有的新式糖廠, 將之合併, 設立台灣糖業公司。¹ 從此開始, 台糖公司是台灣砂糖市場的獨賣者。本文的估算結果與吳聰敏 (1991) 之結果有明顯差異, 特別是在經濟管制及惡性物價膨脹期間。

參考文獻

《中華民國國民所得編算方法說明》(1969), 行政院主計處, 台北。

吳聰敏 (1991), “1910年至1950年台灣地區國內生產毛額之估計”, 《經濟論文叢刊》, 19(2), 127–75。

—— (1997), “1945–1949年國民政府對台灣的經濟政策”, 《經濟論文叢刊》, 25(4), 521–54。

《臺灣省五十一年來統計提要》(1946), 行政長官公署。

A System of National Accounts (1993), Commission of the European Communities, IMF, OECD, and UN, Brussels/Luxembourg, 3rd edn.

¹有關於戰後初期國民政府在台灣的經濟政策之檢討, 請見吳聰敏 (1997)。

```
% exam6.ctx
\documentclass[12pt]{article}
\usepackage{natbib}
\setlength{\bibsep}{2pt}
\renewcommand{\refname}{\ctxfbb      參考文獻}

\title{ 台灣農畜業之生產額：1902--52}
\author{ 吳聰敏}
\date{2002.3.1}
\begin{document}
\maketitle
\fontsize{12}{18pt}\selectfont
根據 \citet[ 頁18--22]{DGBAS69},
農畜業生產額由下列三大項加總而得
(1) 農耕生產總額, (2) 畜禽生產總額, (3) 其他收入。
依 \textit{\citeauthor{SNA93}} (1993, 頁150--52),
產品生產額應以 basic price 乘上數量求算。
本研究農畜業產品生產額資料主要取自 \citet{Stat46} 。
```

日本投降之後，國民政府接收所有的舊式糖廠，將之合併，設立台灣糖業公司。^{\footnote{有關於戰後初期國民政府在台灣的經濟政策之檢討，請見\citet{WuTM97}。}}從此開始，台糖公司是台灣砂糖市場的獨賣者。本文的估算結果與\citet{WuTM91} 之結果有明顯差異，特別是在經濟管制及惡性物價膨脹期間。

```
\bibliographystyle{cw}
\bibliography{test}
\end{document}
```

- 學術論文經常須列舉參考文獻。L^AT_EX 提供 bibtex 之方便排版參考文獻。但是，中文參考文獻之格式與英文不同。故 cwL^AT_EX 提供 cwibibtex 工具程式，以便處理中英文參考文獻。
- 排版參考文獻的方法如下：首先將個人使用的文獻依一定格式輸入於資料檔案中，每一文獻取一名字。本例資料檔名為 test.bib，其內之文獻名字有：SNA93，Stat46 等。
- 文稿內以 \citet 或 \citep 指令引入參考文獻。這兩個指令是 natbib 巨集套件所提供的，故我們須於全文設定區引用該巨集套件。
- 學術期刊各有其排版參考文獻之格式。cwL^AT_EX 提供一個中英文皆適用之格式，取名為 cw.bst。欲使用此格式，我們於文稿末端以 \bibliographystyle 指令引入。排版時，L^AT_EX 先收集文稿內引用之所有文獻，自文獻資料檔案內取出各文獻，再排版於文稿末端。
- 進一步細節請參考 16 章之說明。

國民所得兩萬美元

吳曉敏

台灣的生活水準在提升嗎？若從大街小巷充斥著賣名牌的商店來看，答案是肯定的。但如果從生活環境品質日益惡化來看，答案則是否定的。

三十年前，台灣的所得 (income) 尚低。一般民衆最關心如何提升所得；因此政府施政也以提升所得爲主要目標，這不難理解。但是，所得逐漸上升之後，民衆的偏好、需求會逐漸改變。每一個國家都是如此，台灣也不例外。最簡單的一個例子，幾年來出國旅遊風氣

日盛，這表示民衆對休閒的需求日增。因爲台灣本島可供旅遊休閒的場所太少，大家只好往國外跑。（但是，並不是每一個人都有能力到國外渡假。）

公元兩千年時，台灣每人平均國民所得高達兩萬美元。上下班時，每人身著名牌，在公車或自用車子裡動彈不得，空氣品質及水質比今天更惡化；遇到「假日」除非有錢出國，否則只好呆在客廳裡吹冷氣、看電視。這樣子的兩萬美元所得的意義在那裡呢？

```
% file: exam7.ctx
\documentclass[12pt]{article}
\usepackage{multicol}
\usepackage[landscape,dvips]{geometry}
\columnsep=6mm
\textwidth=18cm
\thispagestyle{empty}

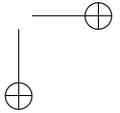
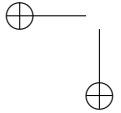
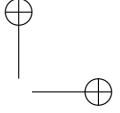
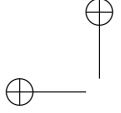
\begin{document}
\fontsize{12}{22pt}\selectfont
\ctxfvm
\hspace*{6mm}
{\large\ctxfvbb 國民所得兩萬美元}
\hspace*{8mm}{\ctxfvf 吳聰敏}

\bigskip
\begin{multicols}{2}
台灣的生活水準在提升嗎？
若從大街小巷充斥著賣名牌的商店來看，
答案是肯定的。
但如果從生活環境品質日益惡化來看，
答案則是否定的。

三十年前，台灣的所得 (income) 尚低。
一般民衆最關心如何提升所得；
因此政府施政也以提升所得為主要目標，
這不難理解。
但是，所得逐漸上升之後，民衆的偏好、需求會逐漸改變。
每一個國家都是如此，台灣也不例外。
最簡單的一個例子，幾年來出國旅遊風氣日盛，
這表示民衆對休閒的需求日增。
因為台灣本島可供旅遊休閒的場所太少，
大家只好往國外跑。
(但是，並不是每一個人都有能力到國外渡假。)

公元兩千年時，台灣每人平均國民所得高達兩萬美元。
...
\end{multicols}
\end{document}
```

- cwTeX 提供5套垂直中文字體。直排文稿仍須考慮頁眉、頁足、標題、與註解等問題。我們並無這方面的排版經驗，故實際之應用尚待進一步測試。
- 直排字體僅提供5種：明體、粗黑體、圓體、楷體、與仿宋體，而且只有常用字。直排字體指令是在橫排字體指令之前加上 v ，例如，直排明體字指令為： \ctxfvm ；楷體字指令為： \ctxfvk 。
- 直排中文字是原中文字轉90度，故紙面可採橫式，使每一行可排入較多字數。在英文中，橫式紙面稱為 landscape 。本例使用 geometry 巨集套件之指令設定橫式紙面。
- 直排中文時，由上而下的一行通常太長，故本例使用 multicol 巨集套件以兩欄式編排。



3 排版方法

個人電腦自 1980 年代初期逐漸普及以來，使用最廣的應用軟體可能是文書排版軟體。文書排版軟體可分為兩大類：幕前排版 (What-You-See-Is-What-You-Get) 與幕後排版。絕大部分的英文文書處理軟體，如 Word 或者 Word Perfect，都是幕前排版軟體， $\text{T}_\text{E}\text{X}$ 則是幕後排版軟體。

使用幕前排版軟體排版很方便。當我們輸入文字時，軟體立即編排版面。最後的列印結果和顯示器所見到的幾乎完全相同。相反的，若使用幕後排版軟體，我們必須先以文字編輯軟體 (editor) 輸入文稿與排版指令，以排版程式編排 (typeset)；再利用預覽軟體 (previewer) 觀看編排結果。若一切無誤，最後由列印機印出。幕後排版之過程間接，但如果你追求的是品質，幕後排版軟體有時是唯一的選擇。在專業排版領域內， $\text{T}_\text{E}\text{X}$ 是一套有名的高品質幕後排版系統。

$\text{T}_\text{E}\text{X}$ 排版軟體可以在不同的作業系統上使用，為了易於辨識，不同作業系統上之 $\text{T}_\text{E}\text{X}$ 系統各有其稱呼。譬如，1990 年代初期的 MSDOS 與 OS/2 作業系統使用者大都使用 $\text{emT}_\text{E}\text{X}$ 。本書主要介紹 $\text{MiK}_\text{T}_\text{E}\text{X}$ 與 $\text{fpT}_\text{E}\text{X}$ 兩套系統，這是針對 Windows 作業系統所發展出來的。

欲使用 $\text{T}_\text{E}\text{X}$ 排版，電腦中須安裝 $\text{MiK}_\text{T}_\text{E}\text{X}$ 或 $\text{fpT}_\text{E}\text{X}$ 軟體。不過， $\text{T}_\text{E}\text{X}$ 是幕後排版，除了安裝 $\text{T}_\text{E}\text{X}$ 之外，我們還需要預覽/列印軟體。Windows 系統下有好幾套預覽/列印軟體，本章將分別介紹 YAP/Windvi, GSview 與 Acrobat Reader 三種。這三種軟體各有其特點，可滿足不同使用者之需求。除了以上軟體之外，我們還需要一套文字編輯軟體以輸入文稿。本章主要介紹 WinEdt，此一軟體與 $\text{T}_\text{E}\text{X}$ 結合得很好；更重要的是，它能辨識中文，輸入中文字時不致出現錯誤。

功能較強的文字編輯軟體都可以讓使用者定義功能鍵。譬如，大部分應用軟體都將 [F1] 功能鍵定義為「功能說明」。使用軟體時若遇到困難，按下 [F1] 鍵，視窗內即出現相關問題之說明。如果事先設定好功能鍵，輸入排版文稿之後，按下功能鍵即可執行排版與預覽/列印動作，不須離開文字編輯軟體視窗。如果你依下一章所介紹的方法成功安裝軟體，安裝程式將在 WinEdt 中設定幾個功能

鍵,讓排版工作變得輕而易舉。以下說明如何透過 WinEdt 之功能鍵或功能圖像 (icon) 進行排版。

3.1 排版步驟

cwTeX 安裝完成之後,硬碟中將新增 c:\texmf 與 c:\localtexmf 兩個檔案夾。在 c:\texmf\cwtex\examples 檔案夾下有好幾個測試檔,其中之一為 test.ctx。以 WinEdt 開啓此檔案,其內容如下:

```
\documentclass[12pt]{article}
\begin{document}
\fontsize{12}{18pt}\selectfont
以 \TeX{} 排版中文很容易;
數學式之排版,如  $\sqrt{\beta}$ , 尤其精確而且簡單。
\end{document}
```

這是一篇簡單的文稿,第一道指令設定文件類別為 article,並選用 12pt 之英數字體;第 2 行指令為 document 指令環境之起頭,指示下一行開始為文稿之內容。第 3 行以 \fontsize 指令選用 12pt 之英數字體,行距重新設定為 18pt。本文未選用中文字體,故將以內定之明體字排版。最後一指令指示文稿結束。

文稿輸入完畢之後,我們如何排版呢?一般而言,使用 cwTeX 系統排版文稿須經過三個步驟:第一步是執行 cwtex 將文稿內之中文字轉換為 TeX 字體指令;第二步是執行 latex 進行排版;最後一步是預覽/列印。這三個步驟可以經由不同途徑完成,每一途徑各有其特點。使用者可依個人偏好,選擇最適當途徑。以下之說明以 test.ctx 測試檔為例。

途徑 1: cwTeX → ETeX → DVIPS → GSview

途徑 1 是最典型的排版方法。在 WinEdt 視窗內輸入文稿或開啓原已輸入完成之檔案,首先按 [F9] 功能鍵將中文字轉換為 TeX 字體指令;接著按下 [F10] 進行排版。以 test.ctx 為例,在 WinEdt 視窗內按下功能鍵 [F9] 即等於是執行下列指令:

```
c:\texmf\cwtex\examples>cwtex -d=c:\xtemp test.ctx
```

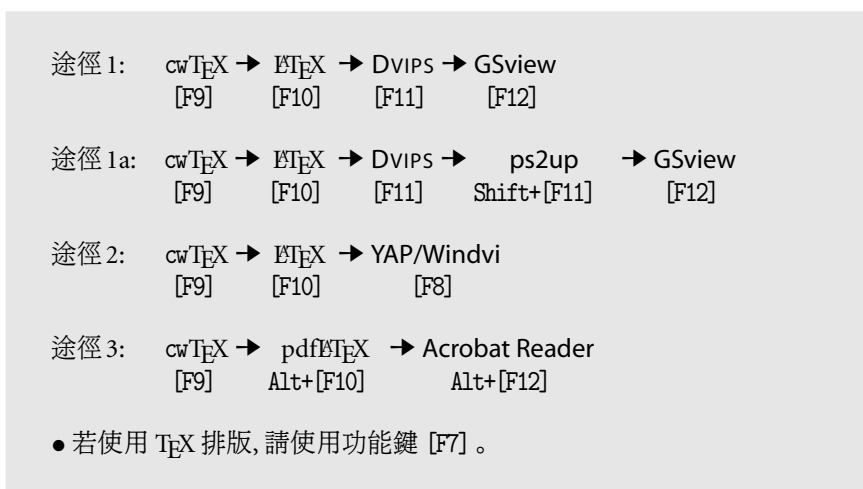



圖 3.1: 排版步驟

執行完畢後, 硬碟內將產生 `test.tex` 與 `cinput.tex` 兩檔案。執行 `cwtex` 時, 可加入選項。上面指令中加入 `-d=c:\xtemp` 選項, 其作用是將程式執行之後所產生之輔助檔案全部移入 `c:\xtemp` 工作檔案夾內。若不加入此一選項, 排版後所有檔案將儲存於 `test.ctx` 所在之檔案夾內。將執行結果移入特定位置之工作檔案夾有其方便之處。因為輔助檔案集中一處, 每隔一段時間清理時, 不須到各檔案夾尋找檔案。

按下功能鍵 [F9] 時, 顯示器畫面即開啓 DOS 視窗, 其內顯示 `cwtex` 之執行訊息。若執行過程未遭遇錯誤, 工作完成之後, DOS 視窗自動關閉。若 `test.ctx` 檔案內之排版指令有錯誤, 其訊息會凍結在視窗內, 等待進一步處理。功能鍵 [F10] 也是以類似的方式運作, 其作用是執行 `latex` 程式進行排版。有些排版之錯誤訊息會在 [F10] 階段才出現, 此時須回到 WinEdt 視窗修改文稿檔案。修改完畢, 關掉 `latex` 視窗, 再按 [F9] 與 [F10] 兩鍵重新排版。

排版之後, 按 [F11] 功能鍵即執行 DVIPS 程式將結果轉換為 PostScript 格式, 檔名為 `test.ps`。預覽/列印時, 須按 [F12] 功能鍵啓動 GSview 軟體。圖 3.2 顯示文字編輯視窗與 GSview 預覽結果之畫面。若文稿須修改, 可跳回文字編輯視窗修改原稿, 之後按 [F9], [F10] 與 [F11] 三功能鍵, 再將滑鼠移回 GSview 視窗, 畫面即更新為修正後的結果。

比起底下介紹的途徑 2, 途徑 1 過程多一步驟, 預覽速度稍慢一些, 但它最大

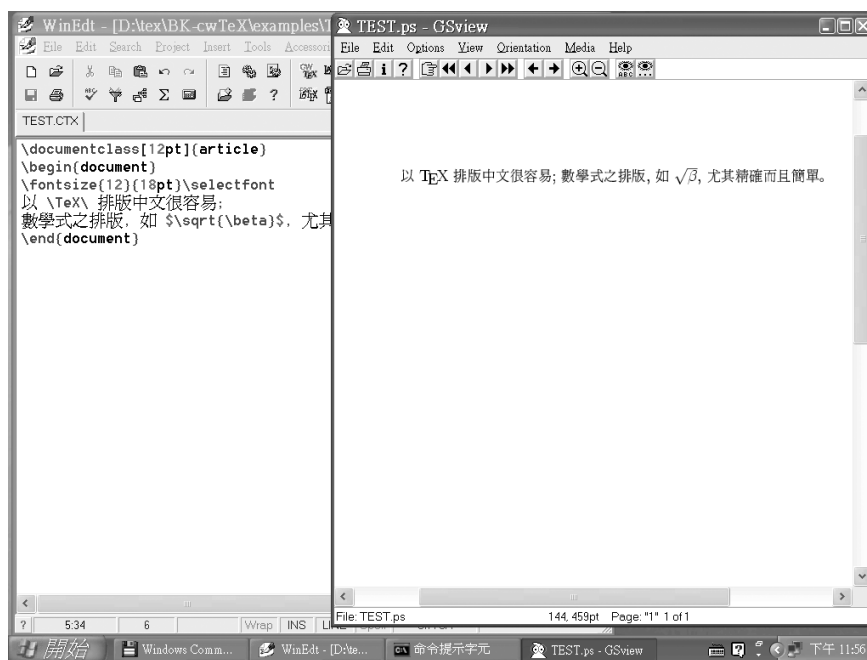


圖 3.2: WinEdt 與 GSview

的優點是直接使用描邊字型, 因此排版結果可以在高解析度之印表機或輸出機上印出, 品質相當完美。此外, 如果文稿中引用了其他軟體繪製之圖形, 則途徑 1 是最佳選擇。一般的繪圖軟體可將圖形檔儲存為多種格式, 其中使用彈性最大, 效果最佳之圖形格式是 PostScript。

底下途徑 2 所將介紹的 YAP 軟體也可以預覽與列印 PostScript 圖形, 但使用的限制較多。故我們建議排版時儘可能使用途徑 1 的方法。有關於引用外製圖形之方法, 請見第 11 章。

途徑 1a: $\text{cwTeX} \rightarrow \text{ETeX} \rightarrow \text{DVIPS} \rightarrow \text{ps2up} \rightarrow \text{GSview}$

途徑 1a 與途徑 1 之唯一差別是在預覽/列印之前執行 `ps2up` 程式對 `test.ps` 檔案作進一步轉換。轉換之後, 每一頁版面會縮小, 然後兩頁合併為一頁列印。在排版術語中, 這稱為 2up, 目的是節省用紙。在 WinEdt 視窗內, 我們只須按下 `Shift+[F11]` 功能鍵, 電腦即自動執行 `\texmf\cwtex` 檔案夾內之 `ps2up` 批次檔。此批次檔之內容如下:

```
ren %1.ps %1.psa
call psnup.exe -2 -pletter -s0.8 %1.psa %1.ps
del %1.psa
```

第2行指令中之 `-s0.8` 指示將文稿縮小為 80%；必要時，可自行調整此數字放大或縮小。若將列印結果再影印一份，可選擇放大 125%，影印結果即為原設計之尺寸。另外，`-pletter` 是設定使用 letter size 紙張，必要時可改用其他尺寸。

欲使用 `ps2up` 程式，文稿前端須加入指令調整版面位置，否則文字版面可能列印到紙張外面。我們的經驗是在文稿前端加入下列兩行指令：

```
\hoffset=-2cm
\voffset=3cm
```

以上說明使用功能鍵排版的方法。除此之外，我們也可以點選 `WinEdt` 視窗內之功能圖像啟動排版程式。如圖 3.3 所示，視窗上方靠近中央的位置有一 `cwTeX` 圖像，以滑鼠點選此圖像，即等於是按下 `[F9]` 功能鍵。其右方的三個圖像分別代表 `TeX`，`DVIPS`，與 `GSview`，而 `DVIPS` 圖像下方為代表 `ps2up` 程式之圖像。

如果要以 `YAP` 預覽/列印，請點選 `GSview` 右方之圖像。在 `MiKTeX` 系統下，此圖像代表 `YAP` 軟體；在 `fpTeX` 系統下，此圖像代表 `Windvi` 軟體。`cwTeX` 圖像下方為代表 `pdfTeX` 之圖像。此一程式可將 `TeX` 檔案排版為 `PDF` 格式，以下將有進一步說明。`pdfTeX` 右方為 `Acrobat Reader`，用於預覽/列印 `PDF` 檔案。

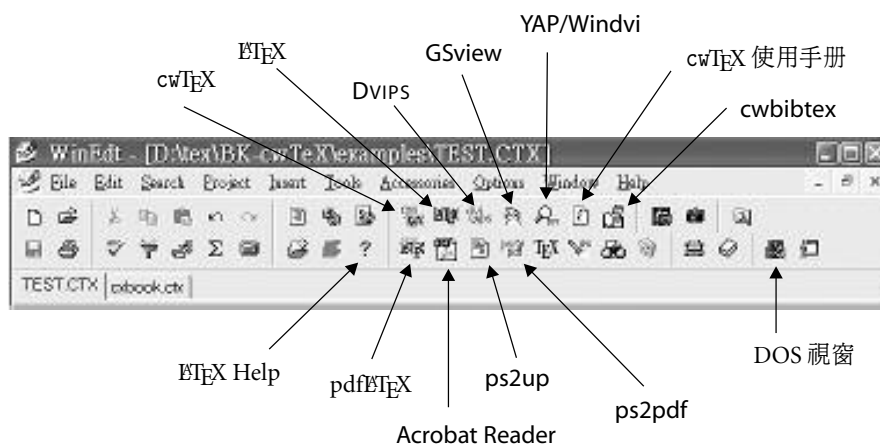


圖 3.3: WinEdt 圖像設定

為方便查詢，我們另外設定兩個說明檔圖像。第一個置於 pdf \LaTeX 左方，其內容是 \LaTeX 指令之簡要說明。第二個是本書全部內容，格式為 PDF。以滑鼠點選 YAP/Windvi 右方之圖像（白紙背景上有一個英文 *i* 字母），電腦將啟動 Acrobat Reader 軟體，並開啓 cw \TeX 使用手冊以供查詢。

途徑2: cw \TeX → \LaTeX → YAP/Windvi

途徑 2 之排版方式與途徑 1 之差別是預覽/列印之方法不同。以 [F10] 功能鍵排版完成之後，按 [F8] 使用 YAP 預覽排版結果。若一切滿意，由 YAP 軟體視窗內可列印結果。YAP 全名為 Yet Another Previewer，這是 MiK \TeX 所提供的軟體。如果你安裝的是 fp \TeX ，[F8] 功能鍵將啟動 Windvi 軟體。

\LaTeX 排版後將產出 test.dvi 檔案，其內並不含有預覽/列印所需之字型檔。以 YAP 預覽/列印時，軟體會自動自硬碟中取得描點字型 (bitmapped font)。譬如，test.ctx 檔案中使用了 CMR12 之英文字體，因此 YAP 須取得 CMR12.PK 之描點字型才能預覽/列印。此測試檔檔案中也使用了明體中文字體 MO，因此軟體也須取得 MO.PK 描點字型。

第一次使用 YAP/Windvi 預覽排版結果時，軟體視窗內會出現訊息，說明正在創造描點字型檔。描點字型產生之後即存放於硬碟內，以供下一次使用。因此，YAP/Windvi 程式使用一段時間之後，硬碟中已累積了相當數量的描點字型。此時預覽/列印排版文稿即不須再產生描點字型，排版結果很快出現在螢幕上。

請特別注意，安裝 MiK \TeX 時，如果選擇 Total 套件組合（參見頁 53），則使用 YAP 軟體預覽時，可能出現某些中文字無法顯示的問題。請試用底下的方法排除問題。首先，請刪除 c:\localtexmf\fonts\source 檔案夾內所有檔案。其次，請刪除 c:\texmf\fonts\source\public\levy 檔案夾內全部的 .mf 檔案，或者移入其他檔案夾內，再重新開啓 YAP 軟體。

途徑3: cw \TeX → pdf \LaTeX → Acrobat Reader

途徑 2 是將排版結果轉換為 DVI 格式，本途徑則是轉換為 PDF 格式。PDF 格式可以說是 PostScript 檔案格式的特別版本，主要特性是具有網路出版的功能，這兩種格式都是 Adobe 公司發展出來的。第 19 章將有進一步說明。文稿 test.ctx 以 [F9] 轉為 test.tex 之後，按下 Alt+[F10] 或者點選 pdf \LaTeX 圖像，即執行

pdf_latex, 排版結果為 test.pdf。預覽/列印 PDF 檔案, 可使用 Acrobat Reader, 功能鍵為 Alt+[F12], 或直接點選功能圖像。

除了將 T_EX 文稿直接排版為 PDF 格式之外, pdf_lT_EX 系統的設計目標之一是要使用 True Type 字體排版文稿, 但這一部份的功能目前仍在發展當中。pdf_lT_EX 可將文稿直接排版為 PDF 格式, 功能甚佳。不過, 受限於 PDF 格式之功能, 此一程式對於文稿內引入外製圖形檔有一些限制。譬如, 文稿內無法直接引用 EPS 圖形檔, 不過可以接受 PDE, JPEG 等圖形檔。15.5.3 節 (頁306) 將介紹 eps2pdf 軟體, 可將 EPS 圖形檔案轉換為 PDF 格式。

不管文稿內有沒有含 EPS 圖形, 要將排版結果轉換成 PDF 格式, 另一個辦法是使用 Ghostscript 所提供之工具程式: ps2pdf。請依途徑 1 先將排版結果轉換為 PostScript 格式, 再點選 ps2pdf 圖像, 工作檔案夾內即產生 PDF 檔案。

3.2 WinEdt 功能簡介

設定功能鍵的目的是為了方便排版工作。cw_lT_EX 安裝程式雖然為 WinEdt 設定一些功能鍵, 但某些情況下使用者可能須再加調整。譬如, 安裝程式假設你是將 GSview 安裝於硬碟 c:。如果 GSview 是安裝於 d:, 功能鍵即無法運作。底下我們將簡單說明如何調整 WinEdt 之功能鍵設定。

表 3.1 列出 WinEdt 軟體之設定。左欄之功能鍵上一節已略有介紹, 其中 [F7] 是用以啟動 tex 程式。如果文稿是以 T_EX 指令排版, 則按 [F9] 之後, 應按 [F7] 排版。表 3.1 右欄是一些常用的編輯指令功能鍵設定。例如, 按 [F2] 鍵即儲存檔案; 功能鍵 [F5] 設定為搜尋字串, [F6] 則為搜尋/替換字串。按下 [F5] 功能鍵之後, 視窗內出現一些選項, 我們可以設定僅於選定區域內尋找字串, 或者搜尋整篇文稿。搜尋時, 我們可設定英文字母是否區分大小寫。功能鍵 Alt+u 設定為「取消上一指令」, 這是所謂的 undo 功能。譬如, 若你上一個動作是刪除一段文字, 但馬上發現不該刪除, 則按下 Alt+u 即可回復原文字。Alt+u 功能鍵可連續使用, 如果連續按鍵 5 次, 則在此之前 5 個編輯動作將依序取消。

WinEdt 軟體視窗內下方工作列上有幾個選項, 右邊第 3 項為 [LINE], 此表示編輯功能是在「文字行」模式。譬如, 如果將游標移於本行之首, 按下 [Shift] 再將游標下移一行, 則本行文字將呈反白。反之, 如果以滑鼠點選 [LINE] 方塊, 編輯功能將進入區域方塊模式 [BLOCK]。在此模式下, 我們可複製或刪除選定之方

表 3.1: WinEdt 功能鍵設定

啓動程式		編輯功能	
[F9]	啓動 <code>cwTeX</code>	[F2]	儲存檔案
[F10]	啓動 <code>latex</code>	[F5]	搜尋字串
[F11]	啓動 <code>DVIPS</code>	[F3]	往下搜尋同一字串
[F12]	啓動 <code>GSview</code>	Shift+[F3]	往上搜尋同一字串
Shift+[F11]	啓動 <code>ps2up</code>	[F6]	搜尋/替換字串
[F8]	啓動 <code>YAP</code>	Ctrl+[F1]	行首加入 % 指令
Alt+[F10]	啓動 <code>pdflatex</code>	Alt+u	取消上一指令 (undo)
Alt+[F12]	啓動 <code>Acrobat Reader</code>	Ctrl+g	游標移至特定行 (goto)
[F7]	啓動 <code>tex</code>	Alt+d	刪除本行
		Alt+k	刪除游標至行尾之文字

塊區域。

視窗下方工作列最右邊一項設定所謂的 Document Mode, 原始內定值為 `TeX`。如果是中文稿件, 我們建議使用 `cwTeX` 模式。點選工作列最右邊之方塊, 即出現一小視窗, 由其中之 Document Mode 可選用 `cwTeX` 模式。WinEdt 文字軟體功能甚多, 自行調整之空間很大。在軟體視窗內按 [F1] 功能鍵即開啓「使用說明」, 有興趣者請自行研究了解。

WinEdt 對大部分常用之編輯指令都設有功能鍵。不過, 每一使用者各有其偏好與習慣。若你不習慣以上之功能鍵設定, 可重新設定。請見 WinEdt 軟體之說明檔。

3.3 排版訊息

執行各排版步驟時, 顯示器上會出現一些訊息。了解這些訊息的意義對於偵測錯誤很有幫助。第一類訊息之內容為記錄處理過程及結果; 第二類訊息則指出錯誤所在。

執行中文轉換程式時, 顯示器上將出現下列訊息:

```
This is cwTeX preprocessor, version 16.5g.
**TEST.CTX
(TEST.CTX [7])
Output written on c:\xtemp\TEST.tex & cinput.tex.
Transcript written on test.xlg.
```

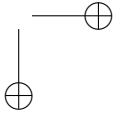
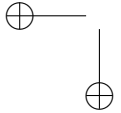
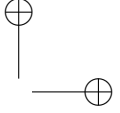
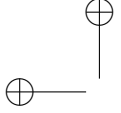
第 1 行訊息末端的 16.5 為 `cwtext` 的版本號碼。第 3 行的 [7] 表示文稿長度為 7 行。第 4 行訊息則說明中文字轉換之後，產生 `test.tex` 與 `cinput.tex` 兩個檔案。這兩個檔案直接移入 `c:\xtemp` 檔案夾內。最後一行的意思是說：中文轉換過程之訊息將記錄於 `test.xlg` 檔案。因此，如果來不及從視窗內讀取所有的訊息，我們仍可事後開啓此一檔案進行了解。

以 `latex` 編排文稿時，也會產生許多訊息。`latex` 程式所處理的並不是原始的 `test.ctx`，而是經 `cwtext` 轉換中文字之後的 `test.tex` 與 `cinput.tex`。執行指令之後，一方面顯示器視窗內出現排版訊息；同時，所有訊息內容也將記錄於 `test.log` 檔案內。以測試檔為例，排版訊息如下：

```
This is TeX, Version 3.14159 (MiKTeX 1.20) (preloaded ...
**TEST
(TEST.tex
LaTeX2e <1998/12/01>
Babel <v3.6k> and hyphenation patterns for american, ...
ohyphenation, loaded.
(cinput.tex) (d:\texmf\tex\latex\base\article.cls
Document Class: article 1999/01/07 v1.4a Standard LaTeX ...
(d:\texmf\tex\latex\base\size12.clo
...
[1] (TEST.aux) )
Output written on TEST.dvi (1 page, 644 bytes).
```

上述訊息最後一行 ... on `TEST.dvi` (1 page, 644 bytes)。說明排版結果長度為 1 頁，儲存於 `test.dvi` 檔案中，檔案大小為 644 bytes。利用此檔案，我們可進一步以軟體預視與列印排版結果。

`TeX` 編排之後的檔案，其附加檔名為 `dvi`，代表 `device independent`，其意義是若將此文稿在不同的列表機 (devices) 上印出，或在不同的顯示器上預視，版面模樣將完全相同，差別只是在於精細程度不同。因此，若我們有一部噴墨印表機，則文件初稿可以先在此印表機上印出。等到最後的版面確定之後，再以高密度的雷射印表機或相紙輸出機 (phototypesetter) 印出。



4 安裝與設定

cwTeX 中文程式必須依附在 $\text{T}_{\text{E}}\text{X}$ 系統上才能作排版工作。目前，幾乎任何作業系統都可以使用 $\text{T}_{\text{E}}\text{X}$ 排版軟體。中文 cwTeX 程式主要是在 Windows 與 Linux 平台上工作。Windows 平台上有好幾套 $\text{T}_{\text{E}}\text{X}$ 系統，其中可免費使用的有 $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 與 $\text{fp}_{\text{T}}\text{E}_{\text{X}}$ 兩套。 $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 的作者為 Christian Schenk，此系統大約在 1998 年秋天漸趨穩定成熟。 $\text{fp}_{\text{T}}\text{E}_{\text{X}}$ 是由 Unix 平台上之 $\text{T}_{\text{E}}\text{XLive}$ 系統衍生而來，作者為 Fabrice Popineau。許多專業的 $\text{T}_{\text{E}}\text{X}$ 使用者都是在 Unix 平台上工作，並發展出種種工具程式出來，因此 $\text{fp}_{\text{T}}\text{E}_{\text{X}}$ 含有較豐富的工具程式。

根據個人平常使用的感覺， $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 系統執行速度較快。在 Windows 作業系統上， $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 使用者可能也比 $\text{fp}_{\text{T}}\text{E}_{\text{X}}$ 使用者為多。但是， $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 系統有一個問題。標準的 $\text{T}_{\text{E}}\text{X}$ 與 $\text{E}_{\text{T}}\text{E}_{\text{X}}$ 程式在編排文稿之後會產生一個 `.log` 檔案，告知排版者共使用了多少電腦記憶體。排版長篇文稿或書籍時，使用之記憶體可能超過 $\text{E}_{\text{T}}\text{E}_{\text{X}}$ 原始記憶體設定之上限，因而出現錯誤。此時，`.log` 檔案內之訊息就很有用，它可告知我們解決問題的方向。但是， $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 之 $\text{E}_{\text{T}}\text{E}_{\text{X}}$ 程式所產生之 `.log` 檔案卻未提供上述訊息，

不過，對一般使用者而言，這個問題可能並不重要。因此，本書之說明主要以 $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 系統為主， $\text{fp}_{\text{T}}\text{E}_{\text{X}}$ 的安裝與使用方法僅簡單提及。事實上，兩套系統的運作方法很類似，熟悉 $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 系統者，要轉換到 $\text{fp}_{\text{T}}\text{E}_{\text{X}}$ 系統並不難。Windows 作業系統之外， cwTeX 系統也可以在 Linux 平台上運作。Linux 平台上的安裝方法請見 4.7 節 (頁 62)。

4.1 準備安裝

$\text{E}_{\text{T}}\text{E}_{\text{X}}$ 系統與一般軟體有一點很不同。一般 Windows 軟體是由單一軟體公司創造出來的，執行時通常並不需要其他軟體配合。但是， $\text{T}_{\text{E}}\text{X}$ 系統排版需要幾套軟體合作。首先，我們要有文字輸入軟體輸入文稿；文稿輸入之後，以 `cwtext` 轉換

成中文,之後再以 latex 程式編排。最後,排版結果以 DVIPS 轉換為 PostScript 檔案格式,再以 Ghostscript 軟體預覽/列印。

以上5個軟體各由不同人所創作,把它們安裝成一套排版系統時,必須考慮各軟體之間的合作關係。這是為何 cwTeX 系統安裝比較麻煩的原因。以下,我們分節說明安裝過程。

4.1.1 取得軟體

安裝軟體的第一步驟是取得所有軟體。cwTeX 手冊所附光碟包含本書所介紹之所有軟體。如果沒有光碟片,你可以從 cwTeX 網站下載。光碟或網站上之軟體大部分可供個人免費使用,但原作者仍有版權;你不可以出售這些軟體牟取利潤。相關的細節,請參考各軟體之版權聲明。各軟體通常會不定期更新版本。因此,網站上所存放者,可能較光碟上之版本為新。

cwTeX 排版系統是由幾套軟體組合而成,主要可分為三部分:

- 文字編輯軟體 WinEdt; 預覽/列印排版結果之軟體, Ghostscript, GSview 與 Acrobat Reader,
- TeX/LaTeX 排版系統: 在 Windows 平台上可選用 MiKTeX 或 fpTeX。本書主要介紹 MiKTeX 系統。
- cwTeX 中文處理系統與中文字型檔。

cwTeX 光碟內含所有軟體。如果你沒有光碟,國內有幾個網站可下載軟體。各國之 TeX 愛好者成立一國際組織,稱為 TUG (TeX Users Group), 網址為:

<http://www.tug.org>

幾乎所有關於 TeX 系統之資訊都可以自此取得;可下載之軟體則儲存於 CTAN (Comprehensive TeX Archive Network)。

國內中研院計算機中心網站蒐集有 cwTeX 系統與 CTAN 軟體:

<ftp://ftp1.sinica.edu.tw/pub2/tex>

中正大學網站有完整的 CTAN 軟體,但並無 cwTeX 軟體。嘉南藥理科技大學網站有 cwTeX 軟體,但無其他 CTAN 軟體。以上各網站都可以由 cwTeX 網站聯結:

<http://ceiba.cc.ntu.edu.tw/tmwu>

cwTeX 光碟有下列檔案夾：

<code>\Adobe</code>	Acrobat Reader 與 PostScript 列印驅動程式，
<code>\contrib</code>	使用者提供之程式，如漢書軟體之介面程式，
<code>\cwtex</code>	cw TeX 中文處理系統，
<code>\cwtex_Linux</code>	Linux 平台之 cwTeX 系統，
<code>\GSview</code>	GSview/Ghostscript 預覽/列印軟體，
<code>\latex2html</code>	LaTeX2HTML 軟體，
<code>\MiKTeX</code>	MiKTeX 排版系統，
<code>\support</code>	附屬軟體，如 fpTeX 安裝程式與 Java 軟體，
<code>\WinCommander</code>	Windows Commander 檔案管理工具軟體，
<code>\WinEdt</code>	WinEdt 文字編輯軟體。

以上之軟體大部分都是免費使用，但請注意，WinEdt 文字編輯軟體為 shareware，你可以免費試用 30 天。若欲繼續使用，須登記付費。你的電腦中可能已有一套文字編輯軟體。即使如此，我們建議你仍選擇安裝 WinEdt。上一章曾說明如何以 WinEdt 軟體進行排版工作。安裝完成之後，試用幾次，然後在你原使用的文字編輯軟體中作類似的設定，以方便排版工作。

除了 WinEdt 之外，新版的 GSview 與 Windows Commander 也都是 shareware。不過，根據我們的使用經驗，GSview 舊 2.7 版的執行速度相當快，而且可免費使用。如果你暫時無意登記付費，不妨使用 2.7 版。Windows Commander 是一個功能甚佳的檔案管理工具，雖然是 shareware，但並無試用期之限制。

4.1.2 移除舊版

如果電腦內原已安裝 cwTeX 第 2 版，請先移除。以下所謂 cwTeX 第 2 版是指 2000 年 3 月出版的 cwTeX 手冊內所附光碟之系統，其中之 MiKTeX 系統是 1.2 版。此版本並不是以一般的 Windows 方法安裝，因此也不能用 Windows 的方法移除。如果電腦內已有 MiKTeX 1.2 版，硬碟內會有 `\texmf`，`\localtexmf`，與 `\emtex` 等 3 個子目錄。為了安全起見，請將前兩個子目錄改名。等新版安裝完成之後，再將這兩個檔案夾悉數刪除。`\emtex` 子目錄可直接刪除。

如果你原來是安裝 $\text{fpT}_\text{E}\text{X}$ 系統，硬碟內會有5個檔案夾：`\texmf`，`\localtexmf`，`\bin\win32`，`\man`，與 `\info`。最後兩個存放說明檔，可直接刪除。前三個請另取新名字。等新版安裝完成之後，再將之刪除。

4.2 安裝工具程式

安裝軟體的順序很重要，其中 $\text{cwT}_\text{E}\text{X}$ 應該是最後安裝，原因是 $\text{cwT}_\text{E}\text{X}$ 安裝程式必須判斷你是使用 $\text{MiK}_\text{T}_\text{E}\text{X}$ 還是 $\text{fpT}_\text{E}\text{X}$ ，系統是安裝於硬碟 `c:` 還是 `d:`。因應不同的設定，`WinEdt` 軟體之設定必須有所調整。如果你沒有先安裝 `WinEdt`，則最後必須自行調整設定。因此，為避免麻煩，請依本章說明之順序安裝。本節說明如何安裝工具軟體，下一節說明安裝 $\text{MiK}_\text{T}_\text{E}\text{X}$ 與 $\text{cwT}_\text{E}\text{X}$ 之方法。

如果你的電腦曾安裝舊版，本節所介紹之工具程式應該已安置於電腦，不須重新安裝。但是，請參見底下 4.2.2 之說明。若是新安裝，你至少應安裝 `WinEdt` 與 `GSview`。

4.2.1 WinEdt

`WinEdt` 有新舊兩版，新版功能較多，但速度慢很多，故我們建議使用舊版 (1.414 版)。進入光碟 `\WinEdt` 檔案夾內，點選 `setup.exe` 即開始安裝。依內定值，`WinEdt` 軟體將安裝於 `c:\Program Files\WinEdt` 檔案夾內，除非有必要，請勿更改。再提醒一次，`WinEdt` 軟體為 shareware，只能免費試用 30 天。

4.2.2 Ghostscript 與 GSview

$\text{T}_\text{E}\text{X}$ 排版結果可轉成 `PostScript` 格式，再以 `GSview` 軟體預覽/列印。`GSview` 是 `Ghostscript` 程式的介面軟體，你必須同時安裝這兩個程式。如果你曾安裝 $\text{cwT}_\text{E}\text{X}$ 第 2 版，則電腦中可能已有 `Ghostscript 6.0` 版及 `GSview 2.8` 版。但是，`GSview 2.8` 版的預覽速度遠較 2.7 版慢，我們建議移除新版，改安裝舊版。

請使用 `Windows` 之「新增或移除程式」分別移除 `Ghostscript 6.0` 版及 `GSview 2.8` 版。之後，直接執行 `\GSview` 檔案夾下之 `gsv27550.exe`，即可將 `Ghostscript 5.50` 版及 `GSview 2.7` 版安裝於電腦中。此外，我們建議再執行 `\GSview` 檔案夾下之 `gs704w32.exe`，安裝 `Ghostscript 7.04` 版。

4.2.3 Acrobat Reader

預覽或列印 PDF 檔案可使用 Acrobat Reader 軟體，這是美國 Adobe 公司所開發，可免費使用。2002 年已有新 5.0 版，但速度較慢。對於預覽 TeX 排版結果而言，新舊版之功能差異不大；若電腦速度不快，我們建議使用 4.05 版。直接執行 \Adobe 子目錄下之 ar405eng.exe，即可安裝。

4.2.4 Windows Commander

你還可選擇安裝 Windows Commander 軟體。此工具程式其實與排版無關，這是一套多功能的檔案管理軟體，可複製/移動檔案，可壓縮/解壓縮，甚至還可以作 ftp，非常好用。直接執行 \WinCommander 子目錄內之 install.exe 即可安裝。

4.3 安裝 MiKTeX 與 cwTeX

在 Windows 平台上，MiKTeX 與 fpTeX 是兩套可免費使用之 TeX 系統。對於一般使用者而言，兩者之差異不大。本節僅介紹安裝 MiKTeX 的方法，如果你要使用 fpTeX 系統，請參考 4.6 節 (頁 61) 之說明。

4.3.1 MiKTeX

光碟 \MiKTeX\setup 檔案夾內有 setup.exe 程式，直接執行程式即開始安裝。安裝時，若不指定，安裝程式將安裝全部套件 Total，但我們也可以選擇 Small 或 Large，僅安裝部分套件。如果選擇 Small，本書所介紹的一些巨集套件將不會安裝進你的電腦，你必須自行加裝。為避免麻煩，如果硬碟空間夠大，建議選擇 Total 組合。

依內定值，系統將安裝於 c:\texmf，個人檔案則安裝於 c:\localtexmf 子目錄。你可以選擇安裝於 d:，e:，或 f: 硬碟，但勿選擇 g: 以後的硬碟，否則底下 cwTeX 安裝程式無法判別。安裝完成之後，若是 Win95/98/ME 系統，必須重新開機；Windows 2000 或 XP 系統則不需要。

新版 MiKTeX 在許多方面都比舊版進步，但你如果是排版長篇書籍，因為使用太多的字型檔，DVIPS 可能無法轉換出 .ps 檔。解決之道是改用 MiKTeX 2.2 測試版之 dvips，安裝方法如下：

1. MiKTeX 程式安裝於 `c:\texmf\miktex\bin` , 請將其內之 `dvips.exe` 改名為 `dvips12.exe` ,
2. MiKTeX 2.2 測試版壓縮檔置於光碟 `\MiKTeX\tn\beta-packages` 檔案夾內, 請將 `miktex-dvips-bin.cab` 壓縮檔內之 `dvips.exe` 解壓於 MiKTeX 程式檔案夾內,
3. 將測試版壓縮檔 `miktex-msvcrt.cab` 內 3 個 `.dll` , 與 `miktex-runtime-bin.cab` 內 9 個 `.dll` 檔案亦解壓於 MiKTeX 程式檔案夾內,
4. 將測試版壓縮檔 `miktex-graphics-bin.cab` 內所有 `.dll` 與 `.exe` 檔案解壓於 MiKTeX 程式檔案夾內。

安裝 MiKTeX 時若選擇 `Small` 或 `Large` 套件組合, 安裝完成之後你可以再加裝新巨集套件或新軟體。欲添加新巨集套件, 請由 Windows 之

開始|程式集 |MiKTeX |MiKTeX Options |Package |Other

點選欲加裝之套件。

請注意, 如果你的作業系統是 Win95 或 Win98, 而且未曾更新, 安裝 MiKTeX 時可能出現錯誤訊息, 說明某些系統檔案找不到或須較新的版本。須更新的系統檔案包括: `comctl32.dll` , `mfc42.dll` , `msvcpl60.dll` , 與 `msvcrt.dll` 。除了以上的情況之外, 如果你的電腦是全新的作業系統, 而且尚未安裝其他軟體, 也可能出現類似錯誤訊息。此時, 你可先執行光碟 `\MiKTeX\download` 檔案夾內之 `vcredist_enu.exe` 軟體進行更新。

4.3.2 cwTeX

光碟 `\cwtex` 檔案夾內有 `setup.exe` 程式, 直接執行即開始安裝。cwTeX 共提供 23 套中文字體, 其中最常用的 5 套稱為基本字體。若硬碟夠大, 你可以選用 `Complete` 選項, 同時安裝所有 23 套中文字型檔。你也可以選擇 `Typical` 選項, 僅安裝基本中文字體; 事後再加裝其他字體。但是, 網路上之版本僅含 5 套基本字體。

安裝程式會自動判斷你是使用 fpTeX 或 MiKTeX 系統, 並調整 WinEdt 之設定, 以方便排版工作。若安裝過程順利, 安裝的最後階段會要求關閉 WinEdt 軟體, 目的是要調整該軟體的設定。若 WinEdt 軟體是在開啓狀態, 請將之關閉。最

後, 安裝程式會詢問是否重新開機, 以使設定生效。重開機之後, 請將 `c:\xtemp` 內之檔案全部清除, 並依下一節的方法測試調整。

如果你一開始選擇 Typical 安裝選項, 稍後欲加裝其他字體, 請光碟重新執行 `setup.exe`, 並依指示進行。請注意, 字型檔加裝完成之後, 請執行下列指令更新檔案資料系統:

```
c:\>initexmf -u
```

否則更新無效。

從 2.1 版開始, `cwTeX` 是以 Windows 安裝程式安裝。若欲移除 `cwTeX`, 請按 Windows 的標準程序。目前, `MiKTeX` 是 2.1 版, 但隨時有更新程式出現, 未來 `MiKTeX` 2.2 版正式出版之後, 若要升級, 請依下列步驟:

1. 移除 `cwTeX`,
2. 移除舊版 `MiKTeX`,
3. 安裝新版 `MiKTeX`,
4. 安裝 `cwTeX`。

若你安裝的是自網路下載之版本, 其中僅含 5 套中文字體。如果你有 `cwTeX` 第 2 版光碟, 你可以將光碟內之中文字型安裝於 2.1 版系統內。首先, 將 `cwTeX` 第 2 版光碟置於光碟機內。進入 DOS 視窗, 並移入 `c:\texmf\cwtex\util` 檔案夾內。若光碟機為 `e:`, 執行

```
c:\texmf\cwtex\util>addfnt2 e
```

即可加裝一般中文字型。若要加裝垂直中文字型, 請執行

```
c:\texmf\cwtex\util>addfntv2 e
```

請注意, `cwTeX` 安裝程式會更動 `WinEdt` 軟體之設定。在 `\Program Files\WinEdt` 檔案夾內有 `WinEdt.ini` 與 `WinEdt.btn` 兩設定檔, 安裝程式會將原檔案分別備份為: `WinEdt.ini.bak` 與 `WinEdt.btn.bak`, 若有必要, 你可以自行恢復原始設定。

4.4 測試與調整

安裝之後，應測試是否運作正常。在 MiKTeX 系統下，測試文稿 `test.ctx` 置於 `c:\texmf\cwtex\examples` 檔案夾內。若使用 fpTeX，測試文稿則置於

```
c:\Program Files\TEXLIVE\texmf\cwtex\examples
```

檔案夾。啓動 WinEdt 軟體，開啓 `test.ctx` 測試檔，以滑鼠點選視窗上方工作列上之 `cwTeX` 圖像 (icon) (或直接按 [F9])，電腦即執行 `cwtex` 將測試檔之中文字轉換為 TeX 之控制碼。轉換中文之後，以滑鼠點選 `ETeX` 圖像 (或按 [F10])，即啓動排版程式。接下來，點選 `DVIPS` 圖像 (或按 [F11])，排版結果將轉換為 PostScript 格式。最後，點選 `GSview` 圖像 (或按 [F12])，即啓動預覽/列印軟體。

我們也可以使用 YAP 軟體預覽/列印排版結果，這是 MiKTeX 之預覽軟體。(fpTeX 之預覽軟體稱爲 Windvi。)點選 WinEdt 視窗上方工作列上之 `DVI` 圖像即可執行。請注意，安裝 MiKTeX 時，如果選擇 Total 套件組合 (參見頁 53)，則使用 YAP 軟體預覽時，可能出現某些中文字無法顯示的問題。請試用底下的方法排除問題。首先，請刪除 `c:\localtexmf\fonts\source` 檔案夾內所有檔案。其次，請刪移除 `c:\texmf\fonts\source\public\levy` 檔案夾內全部的 `.mf` 檔案，或者移入其他檔案夾內，重新開啓 YAP 軟體預覽。

除此之外，我們還可以使用 `pdfETeX` 將文稿直接排版爲 PDF 格式，再以 Acrobat Reader 預覽或列印；請參見上一章之說明。

4.4.1 調整顯示密度

大部分 Windows 桌面之顯示密度都設定爲 800×600 ，或者 1024×768 。若是後者，WinEdt 之內設值應該是令人滿意的結果。如果 Windows 桌面顯示密度爲 800×600 ，你可以考慮將桌面顯示密度提高；或者將 WinEdt 視窗之設定調整一下。由視窗上方工作列開啓“Options|Preferences|Font”，右下方四個數字使用於調整行距與字距，原先數字爲“16, 20, 2, 10”；請試改爲“16, 16, 1, 8”。視窗左上方可供設定字體，我們的建議如下：先點選 Font，選取「細明體」，按「確定」之後；再度點選 Font，選取 Fixedsys，再按「確定」。

爲得到較好的顯示效果，我們須調整 GSview 顯示密度之設定，方法如下：點選 GSview 視窗上方之“Media|Display settings”，將 Text Alpha 之值由 1 改爲 4，

Graphics Alpha 之值亦由 1 改為 4; 顯示畫面會變得比較精細。

4.4.2 選擇印表機

GSview 軟體須設定印表機。若是 GSview 2.7 版, 請由軟體視窗之

File|Print|Printer Setup |Device

選擇印表機。若是 HP 雷射印表機, Device 請選用 ljet4。實際列印時, 可由 Resolution 選擇列印密度; 由 Pages 選擇列印之範圍。紙張種類可以由 Media 選單內選擇。

萬一電腦的印表機與 GSview 表單上所列出者毫無相似之處, 請試在 Device 空格處選用 mswinpr2 驅動程式。此一驅動程式將列印工作直接交由 Windows 之印表機處理, 列印速度可能慢一些, 但任何印表機應該都可以列印。譬如, 如果你有 HP1200 印表機 (解析度達 1200dpi), 即可透過此一方式列印。

若是安裝 GSview 3.x 或 4.x 新版, 而且使用 HP 雷射印表機, 請由 GSview 軟體視窗之

File|Print|Print Method|Ghostscript Device

選用 ljet4。或者, 你也可以點選 Print Method 下之 Windows GDP Printer。此種設定是透過 Windows 驅動程式列印。

4.4.3 檔案搜尋

排版時, 在 WinEdt 視窗內按 [F9] 功能鍵之後, cwtex 會將 test.ctx 轉換為 test.tex, 並將後一檔案移入 c:\xtemp 工作檔案夾內。接下來按 [F10] 功能鍵時, latex 會在工作檔案夾內排版, 結果則存為 test.dvi。設定工作檔案夾的好處是, 排版過程中所產生的暫存檔全部集中於此。排版完成之後, 只要保存原始 test.ctx 檔案, 其餘暫存檔並無保存之必要。因此, 每隔一段時間, 我們可以將 c:\xtemp 內之檔案全部刪除。

排版文稿內可能含有其他檔案。舉例言之, 第 2 章 examp2.ctx 例子引用了一外製圖形 examp.eps, 排版程式若找不到圖形檔, 即出現錯誤訊息。如果只是偶爾使用圖形, 最簡單的方法是將圖形檔複製一份於 c:\xtemp 工作檔案夾內, 排

版程式一定找得到。但如果引用許多外製圖形,每一圖形檔都要複製於工作檔案夾內並不方便。因此,最好是設定讓排版/列印程式能搜尋圖形檔。

cwTeX 安裝程式會設定讓各程式在 $c:\text{tex}$ 及其下子目錄搜尋檔案。簡單來說,我們設想使用者將個人文稿置於 $c:\text{tex}$ 檔案夾下。假設你正在寫作2篇論文,第一篇與經濟衰退有關,第二篇與經濟成長有關。則兩篇文稿及附屬圖形檔可分別置於 $c:\text{tex}\backslash\text{recession}$ 與 $c:\text{tex}\backslash\text{growth}$ 兩個子目錄內。排版時各軟體可自動搜尋,找到所需的檔案。如果你希望把文稿與圖形檔置於 $d:\text{tex}$ 之下,則須調整設定。

MiKTeX 軟體是依 miktex.ini 之設定搜尋檔案。若系統安裝於硬碟 $c:$,此檔案置於 $c:\backslash\text{localtexmf}\backslash\text{miktex}\backslash\text{config}$ 子目錄內。以文字編輯軟體開啓此一檔案,可發現其內有許多分區,每一分區都有標題。例如, $[\text{LaTeX}]$ 標題下有下列設定:

```
Input Dirs=.;%R\text\latex//;%R\text...;%R\text//;%R\cwtex//;c:\tex//
```

其中,等號右邊第一個句點表示工作子目錄,接下來的分號區隔下一個設定。 $\%R$ 代表 $c:\text{texmf}$ 與 $c:\backslash\text{localtexmf}$ 兩子目錄。最後的 $c:\text{tex} //$ 表示搜尋 $c:\text{tex}$ 及其下所有的子目錄。若個人之文稿檔案及圖形檔都置於 $d:\text{tex}$ 之下,則將所有的 $c:\text{tex} //$ 改為 $d:\text{tex} //$ 即可。

4.5 常見的安裝問題

相較於工具程式與 $\text{T}_{\text{E}}\text{X}$ 系統, cwTeX 之安裝過程似乎較容易出現問題。本小節列舉常見的安裝問題及解決之道:

- 按 $[\text{F9}]$ 出現 “Cannot Run: $\text{cwtex.exe} -d=c:\backslash\text{xtemp} \dots$ ” 訊息

以上訊息通常是指電腦找不到 cwtex.exe 檔案。請依底下步驟作測試。首先,將 $c:\backslash\text{texmf}\backslash\text{cwtex}\backslash\text{examples}\backslash\text{test.ctx}$ 測試檔案複製於 $c:\backslash\text{xtemp}$ 檔案夾內,進入 DOS 模式,執行底下指令:

```
c:\xtemp>cwtex test
```

若出現同樣錯誤訊息,表示 $c:\backslash\text{autoexec.bat}$ 之 set path= 指令設定錯誤,請依下一點之說明修正。反之,若 cwtex.exe 可順利執行,請再執行

```
c:\xtemp>latex test
```

若以上兩步驟都能順利執行，表示排版程式已正確複製於硬碟內，問題是在於 WinEdt 軟體功能鍵未能正確設定。請依底下之說明設定功能鍵。

- 執行程式搜尋路徑

排版時，我們首先須執行 `cwTeX.exe` 程式，此程式置放於 `c:\texmf\cwTeX` 檔案夾下。如果我們是在工作子目錄 `c:\xtemp` 內執行程式，則必須設定讓 Windows 自動找到執行程式，否則會出現錯誤訊息。Windows 各版設定搜尋執行程式的方法並不完全相同，最簡單的方法是在 `c:\autoexec.bat` 檔案內設定 `set path=` 指令。如果是 Win98 以上之系統，你也可以由：

控制台 | 系統 | 進階 | 環境變數 | 系統變數

直接修改。

cwTeX 安裝程式會自動修改 `autoexec.bat`，以設定執行檔搜尋路徑。但有時候電腦原先的設定不正確，以致影響運作。如果你必須自行修正，請依底下說明。簡單來說，如果安裝 MiKTeX，我們必須在 `c:\autoexec.bat` 檔案內最後一行下面加上一行指令：

```
set path=c:\texmf\cwTeX;%path%
```

如果是安裝 fpTeX 系統，則須加上：

```
set path=c:\Progra~1\TEXLIVE\texmf\cwTeX;%path%
```

以上設定有幾個應注意之處。首先，Win95/98/ME 之路徑不能超過 8 個字母，故我們必須將長檔名 `\Program Files` 改為短檔名 `\Progra~1`。其次，如果 `autoexec.bat` 內有兩行 `set path=` 指令，Windows 僅讀取後一行之設定，前一行之設定變成無效，因此，我們必須把所有的路徑設定於同一行內。但是，這又產生另一個問題。在 Windows 系統上，`autoexec.bat` 檔案內任何一行指令之長度不能超過 128 個字母。如果 `set path=` 指令行中加入太多路徑設定，該行指令長度可能超過 128 個字，而變成無效。

解決的辦法是如上例所示，在指令末端加上 `;%path%`。其中，分號是區隔前後兩個設定，而 `%path%` 則是將前面的路徑設定加入本行設定末端。如此一來，即不必擔心 `set path=` 指令行太長的問題。最後要注意的是，Windows ME, Win2000 或者 Windows XP 可能會把 `autoexec.bat` 變為隱藏檔。此時，安裝程式可能無法作正確設定，必須自行手動調整。

欲了解設定是否正確，最簡單的方法是進入 DOS 模型，執行：

```
c:\>path
```

畫面上會出現作業系統目前的路徑設定，請仔細檢查是否有錯誤。其中，各路徑之間應以分號區隔，此外不應有其他的文字或指令。

- 排版時出現 “Can’t find yap.exe(MiKTeX) or windvi.exe(fpTeX) ” 訊息
`cwTeX` 執行時會先確定電腦內安裝的是 `MiKTeX` 或者 `fpTeX`。如果你已安裝以上系統之一，則出現以上訊息的原因可能是作業系統未正確運作。作業系統未能正確運作，原因很多，可能是你的防毒軟體運作不正常，也可能是作業系統安裝有問題。舉例來說，你如果從 Windows ME 直接升級到 Win2000，而不是全新安裝 (clean install)，新的作業系統的運作可能就會有問題。
- 排版時出現 “I can’t find m0.tfm ” 訊息
 可能原因有二：第一是字型檔案未複製於硬碟內，第二是未正確設立字型對應 (font mapping) 檔案。中文字型規格檔案置於下列檔案夾：

```
c:\texmf\fonts\tfm\cwTeX
```

請檢查檔案夾內是否有 `m0.tfm` 字型規格檔。若字型檔存在，則問題是在於字型對應檔。中文字型對應檔有兩個，一般正常字型對應檔名為 `cwTeX.map`，變形字對應檔為 `cwTeX1.map`，應置於 `c:\texmf\dvips\cwTeX` 檔案夾內。若硬碟尚安裝直排字型，其對應檔名為 `cwTeX2.map`。

- 設定 WinEdt 功能鍵
 如果安裝 `cwTeX` 系統之前沒有先安裝 WinEdt，則其功能鍵須自行設定。使用 `MiKTeX` 者，請將 `\texmf\cwTeX\util` 檔案夾內之 `WinEdt.ini` 與 `WinEdt.`

btn 複製於 c:\Program Files\WinEdt 子目錄內。另外, cwTeX.bmp 則移入 c:\Program Files\WinEdt\Bitmaps\Buttons\ 子目錄內。複製之前, 請先關閉 WinEdt 軟體。

若使用 fpTeX, WinEdt.ini 等檔案置於

```
c:\Program Files\TEXLIVE\texmf\cwtex\util
```

子目錄內, 請依同法處理。

- 順利使用一段時間之後, 突然無法執行

可能原因是 WinEdt 之設定檔被改掉。WinEdt 是 shareware, 安裝之後可使用約一個月。過了期限未註冊, 軟體會自動回復原始的設定, 因此無法以 [F9] 功能鍵執行 cwtex。請將 c:\texmf\cwtex\util 檔案夾內之 WinEdt.ini 覆蓋 c:\Program Files\WinEdt 檔案夾內之同名檔案。前一檔案內已設定執行排版程式之功能鍵。若欲長期使用 WinEdt 軟體, 請註冊。

- 安裝時出現 “... out of environment space ” 訊息

請在 c: 根目錄的 config.sys 檔案中加入下面一行指令:

```
shell=c:\command.com /e:2048 /p
```

重開機, 再重新安裝。

4.6 其他軟體

以上所介紹的是基本系統所需之軟體。但是, WinEdt 與 GSview 事實上都有較新之版本, 而 MiKTeX 也可以用 fpTeX 替代。我們將新版本置於 \support 檔案夾內, 以方便使用者。檔案夾內有 hlatex.zip, 這是早期 emTeX 系統所提供的 htex386, 對於編排複雜的長篇文稿可能有幫助。

如果你不想用 MiKTeX, 而要使用 fpTeX 系統, 底下的資訊或許有幫助。

4.6.1 fpTeX 系統

由 cwTeX 光碟之 \support\fpTeX\0.5\ 檔案夾, 執行 TeXSetup.exe 即開始安裝程序。(檔案夾內 TeXSetup-20020227.exe 為較新版本。) 因為容量限制, 光碟內

並未含 $\text{fp}\text{T}_\text{E}\text{X}$ 系統檔案, 故你須邊下載邊安裝。請注意, $\text{fp}\text{T}_\text{E}\text{X}$ 系統很龐大, 故下載檔案須耗費相當時間。

依內定值, 系統將安裝於 `c:\Program Files\TEXLIVE` 之下, 你可以選擇安裝於 `d:`, `e:`, 或 `f:` 硬碟, 但勿選擇 `g:` 以後的硬碟, 否則底下 $\text{cw}\text{T}_\text{E}\text{X}$ 安裝程式無法判別。安裝程式會選擇基本的程式與巨集套件, 你可以選擇加裝 `tex-psutils` 及 `tex-fontbin` 兩套軟體。若安裝時不加選, 你也可以在稍後實際用到時再加裝, 底下將說明加裝的方法。安裝完成之後, 若是 Win95/98/ME 系統, 你須重新開機, Windows 2000 或 XP 系統則無必要。

安裝完成之後, 你可以再加裝新巨集套件或新軟體。欲添加新巨集套件, 請由 Windows 之

開始|程式集 |TeXLive |Maintenance |Add TeX Package

點選欲安裝之套件即可。

4.6.2 檔案搜尋

$\text{fp}\text{T}_\text{E}\text{X}$ 系統與 $\text{MiK}\text{T}_\text{E}\text{X}$ 一樣, 內含許許多多的檔案。爲了讓排版程式很快找到所需的檔案, $\text{fp}\text{T}_\text{E}\text{X}$ 以 `texmf.cnf` 設定搜尋檔案順序, 檔案置於下列檔案夾內:

```
c:\Program Files\TEXLIVE\texmf-var\web2c
```

假設個人之所有文稿置於 `d:\tex` 檔案夾下, 欲設定排版程式自動搜尋路徑 `d:\tex`, 請將 `texmf.cnf` 內所有的 `c:/tex` 改爲 `d:/tex`。請注意, 子目錄是以右斜線命名, 而非左斜線。

4.7 Linux 版本安裝方法

Linux 平台上之 $\text{cw}\text{T}_\text{E}\text{X}$ 系統是由翁鴻翎先生所創作, 我們還要感謝李果正先生熱心幫忙測試, 以及提供寶貴的意見。

此版本是與 Windows 版本具有同樣的功能, 舉凡 Windows 版本有的程式, 這個版本都有。包括了: `cwtex`, `cwbibtex`, `cwmkidx`, `cwpt1`, 與 `cwttf` 等。所有的測試檔也與 Windows 版本一樣。

4.7.1 安裝程式及字型

`\cwtex_Linux` 檔案夾內有4個檔案: `cwsetup` , `cwtexmain.tgz` , `cwtexbin.tgz` , 與 `allsetup` 。請放在同一個目錄下, 下指令

```
./allsetup
```

就可以安裝完成。

`cwTeX` 共提供23套中文字體, 但其中最常用的5套稱為基本字體。網路上僅提供5套基本中文字體。如果你有 `cwTeX` 第2版光碟, 你可以將光碟內之中文字型安裝於系統內。安裝字形的程式是 `addfont`, 你首先下

```
addfont
```

程式會告訴你有多少的 `cwTeX` 字形可以安裝。假設你想裝黑體 `b` 字形, 請下指令

```
addfont b
```

這時候, 由於程式不知道要安裝的檔案在那裡, 所以會告訴你需要那一些檔案, 然後請你把 `cwTeX` 第2版光碟對應的檔案複製到任一個臨時目錄, 然後再下

```
addfont b
```

就安裝完成。程式本身會檢查是否安裝過某一種字形, 不會重覆安裝, 以免浪費時間。

4.7.2 測試

假如 `texmf` 是在 `/usr/share/texmf` , 則安裝程式會建立

```
/usr/share/texmf/cwtex
```

子目錄, `cwtex` 子目錄下還有

```
/usr/share/texmf/cwtex/examples  
/usr/share/texmf/cwtex/doc  
/usr/share/texmf/cwtex/util  
/usr/share/texmf/cwtex/help
```

其中 `examples` 檔案夾內置放所有 `cwTeX` 的測試檔案。`util` 則放著 `cwpt1` 和 `cwttf` 兩套由 TrueType 字形產生符合 `cwTeX` 規格的程式組, 詳情請見 `ttf2cw.pdf` 的說明。

4.7.3 Linux 與 Windows 版本之差異

由於所有的程式都是與 Windows 同一個版本, 所以使用的方法請見使用手冊和 `cwTeX` 的所有說明檔。不過還是有幾點因為平台差異要特別注意:

- Win/Dos 的檔案拿到 Linux 平台時, 請務必轉換成 Unix 格式, 否則會有錯誤產生。
- 若文稿名為 `my.ctx`, 在 Linux 平台上, 使用 `DVIPS` 請下指令

```
dvips -o my.ps my
```

而不要直接下

```
dvips my
```

否則 `DVIPS` 會把排版結果丟到印表機。

- Linux 的 `dvi` 預視軟體是 `XDVI`。排版之後請在 `Xwindow` 下鍵入指令

```
xdvi my
```

就可以看到排版結果。

4.8 其他作業系統之版本

除了 Windows 與 Linux 作業系統之版本外, 目前在測試中的尚有 FreeBSD 及 OS2 平台之版本。未來也可能有其他平台的 `cwTeX` 出現, 相關的訊息與安裝方法將於 `cwTeX` 網站上宣告。

5 指令與文字

TeX 是幕後排版系統，所有的排版指令必須與文稿內容同時輸入於檔案內。換言之，我們在文稿內所輸入的中英文字，若不是文件內容，就是排版指令。文稿版面是以排版指令編排，排版指令可以設定版面大小、排版表格、設計標題等等。有時候，我們也使用指令**變更字體**以強調某一句話，或者排版數學符號 $\sqrt{2}$ 。

TeX 有極為完整的排版指令，但如果不熟悉指令，使用較困難。本書主要介紹 L^AT_EX，這是由 TeX 衍生出來一套系統，目前也是使用者最多的一套系統。從本章開始，我們將陸續介紹 L^AT_EX 排版指令。有關於 L^AT_EX 系統的介紹可參見 [Lamport \(1994\)](#)，這是 L^AT_EX 軟體原作者所寫的。另外，[Kopka and Daly \(1995\)](#) 是一本入門介紹，內容完整詳盡。[Goossens et al. \(1994\)](#) 則介紹各種巨集套件，對於進階使用者而言，是不可或缺的參考書。本章首先說明輸入文字及排版指令的一般原則。

5.1 符號與註銷指令

文稿內容的輸入很簡單，英文或數字可直接鍵入，中文則須以中文輸入法輸入。但是，有些符號在鍵盤上並沒有對應的字鍵。譬如，商標符號 © 或者英鎊符號 £ 在鍵盤上就沒有對應的字鍵。一般的文書處理軟體通常是以內碼輸入法鍵入特殊符號。但是，在 L^AT_EX 中這些符號是以指令排版。譬如，要排版 £ 記號，我們應鍵入 `\pounds`；要排版商標符號，我們應鍵入 `\copyright`。L^AT_EX 的排版指令絕大部分是以反斜線開頭，英鎊符號用 `\pounds` 來表示，就是一個例子。L^AT_EX 排版數學式子的能力特別強，大部分的數學符號也都是用指令來代表。

L^AT_EX 指令由英文字母組成，不能使用數字；而且大小寫字母是有分別的。譬如，要排版 L^AT_EX 這幾個字，指令是 `\LaTeX{}` 。其中，字母大小寫不可弄錯。如果輸入 `\LaTeX{}` ，將會出現錯誤。

你可能注意到 `\LaTeX{}` 指令末端加上左右大括號。為什麼呢？L^AT_EX 排版時，必須分辨那些是文字，那些是指令。指令是以反斜線 \ 起頭，判斷上不困難。

表 5.1: 重音符號

ò	\'o}	ö	\~o}	ö	\v{o}	ø	\c{o}
ó	\'o}	õ	\=o}	ö	\H{o}	ø	\d{o}
ô	\^o}	ô	\.o}	ö	\t{oo}	ø	\b{o}
ö	\"o}	ö	\u{o}				

但指令結尾如何判斷呢? 原則上, 從指令的第一個字母開始, 到第一個非字母的字元為止, 就構成指令。非字母字元包括空白、標點符號、數字等。因此, 若輸入 `the \LaTeX logo`, 排版結果為: `the \LaTeX logo`。標誌與其後的文字之間未隔開, 原因是指令後面的空白被當成是指令的結束, 因此 `logo` 即緊接著編排。為了避免此種錯誤, 我們應輸入 `the \LaTeX{} logo`, 連續的左右大括號表示指令結束, 緊接其後的空白才會排出空格。另一種輸入方法是 `the {\LaTeX} logo`, 右大括號表示指令結束。第3種輸入指令的方法是 `the \LaTeX\ logo`, 其中反斜線加空格是 `\LaTeX` 排版空格的指令。

`\LaTeX` 提供許多符號可供排版德文、法文或其它語文。表 5.1 列出重音符號 (accents) 及其排版指令。要排版 `ö` 符號, 我們應輸入 `\~o}`。此一指令把一波浪符號加在字母 `o` 的上面。為了易於分辨起見, 字母以大括號括起來。如果大括號中只有一個字母, 則括號可以省略。譬如, 直接輸入 `\~o` 也可得到結果。

表 5.1 的例子是把重音加在小寫字母 `o` 上頭, 當然這些重音可以加在任何字母或符號上面。例如, `\v{z}` 指令可得到 `ẑ`, 而 `\={A}`, 得到 `Ā`。除了重音符號之外, 其它的特殊符號列於表 5.2。利用表中的符號, 我們可以排版德文句子:

Die Höhe der Steuer muß sich durch die Wertung ...	Die H\^ohe der Steuer mu\ss{} sich durch die Wertung ...
----------------------------------------------------	----------------------------------------------------------

事實上, 若要排版長篇德文或法文稿, 應使用 `babel` 巨集套件, 請參見 5.4 節之說明。

文稿內經常須記錄排版當天的日期與文稿檔名。欲排版當天日期, 可使用 `\today` 指令。在文稿內任何地方鍵入此一指令, 排版後即出現當天的日期, 但月份是以英文字排出。譬如:

今天是 June 16, 2002.	今天是 \today.
--------------------	-------------

表 5.2: 特殊符號

§ \S	ı !‘	ı ?‘	† \dag	‡ \ddag
¶ \P	ø \o	Ø \O	ı \l	Ł \L
æ \ae	å \aa	ß \ss	Æ \AE	Å \AA
œ \oe	Œ \OE	© \copyright	£ \pounds	

圖 15.1 (頁288) 的例子將說明如何下指令使日期以中文形式排版。

欲記錄文稿檔名, 可使用 `\jobname` 指令。此項指令僅記錄主檔名, 附加檔名須自行填入。譬如, 排版本書之主檔名為 `cxbook.ctx`, 我們在文稿末端以下列指令記錄檔名與排版日期:

```
r1.ctx (June 16, 2002)           \jobname.ctx (\today)
```

輸入文稿時, 有時候我們必須在某處加註註銷說明。這些說明並不是排版文稿的一部分, 只是作提醒或補充說明之用。此時, 我們應使用 `%` (註銷指令, `comments`) 指令。第 2 章例 1 的首行為:

```
% file: exampl1.ctx
```

這是用以註明檔案名稱。因為其開頭有 `%` 指令, 排版時 `%` 符號及其後的文字與指令將完全略過。TeX 的指令大都是以反斜線起頭, 但註銷指令是一個例外。

若註銷指令出現在一行文字中間, 指令之前的文字會出現, 但指令本身連同後面所有的文字都不會排版出來。因此, 你可能馬上想到一個問題: 如何排版百分比 `%` 符號? 底下是一個例子:

```
台灣政府支出占 GDP 比率約 27和      台灣政府支出占 GDP 比率約 27%,
其他國家相差不大。                  和其他國家相差不大。
```

因為數目字 27 之後緊接著註銷指令, 其後的逗號即略過不處理。要排版百分比 `%` 符號, 我們必須輸入 `\%`。因此, 正確的指令及排版結果如下:

```
台灣政府支出占 GDP 比率約 27%,      台灣政府支出占 GDP 比率約 27%,
和其他國家相差不大。                和其他國家相差不大。
```

表 5.3: 以指令方式輸入之英數符號

符號	符號輸入指令	功能
%	\%	註銷
\	<code>\backslash</code>	定義指令
{	<code>\{</code>	左大括號
}	<code>\}</code>	右大括號
&	\&	表格排列
\$	\\$	數式模式
#	\#	記錄巨集指令參數
~	<code>\~{}</code>	加入空白或防止分割字串
^	<code>\^{}</code>	數式上標
_	<code>_{} </code>	數式下標

若欲排版第 1 欄之符號, 我們必須輸入第 2 欄所示之指令。例如, 要排版百分比符號, 輸入指令為 `\%`。第 3 欄說明第 1 欄符號本身在 \TeX 中之特殊用途。

除了註銷指令 `%` 外, 另外有 9 個符號在 \TeX 中各有其特殊用途。因此, 符號本身也須以指令方式輸入, 我們將這些符號列於表 5.3, 以供參考。除了 `%` 及 `\` 之外, 最常用的符號是左右大括號, 它們是用來界定指令的範圍。譬如, 上面所介紹的重音符號, 即以大括號標示字母範圍。其餘各符號的用途, 我們將陸續介紹。

5.2 英文稿件輸入原則

\TeX 是幕後排版, 版面的控制由指令為之。輸入文字時, 我們只須顧慮文字與指令是否正確, 不必考慮輸入之文稿是否排列整齊。文稿排版是 \TeX 的工作, 不管輸入文稿在顯示器上看來有多雜亂, 只要指令正確, 最後的結果一定令人滿意。有人在輸入文稿時刻意排得整齊, 這是不必要的。

輸入文字時要注意一些基本原則。首先, 在英文的段落之間, 就像在打字機打字時一樣, 單字之間必須留有空白。用打字機打字時, 兩個字中間如果鍵入兩個空白鍵, 紙上會出現連續兩個空白。但在輸入 \TeX 文稿時, 多個空白和一個空白的的作用完全相同。換句話說, 碰到多個空白時, \TeX 會自動將它們刪減成一個空白。另外, 新的一行開頭之處, 不須再留一空白。

由圖 5.1 例子可知, 不管輸入多少空白鍵, 其作用和一個空白鍵相同。輸入時第 2 行的 `by` 和 `spaces` 兩個單字之間雖留有三個空白, 排版之後其間距和正常間

The ends of words and sentences are marked by spaces. It doesn't matter how many spaces you type; one is good as 100.

Note also that one or more blank lines denote the end of a paragraph.

The ends of words and sentences are marked by spaces. It doesn't matter how many spaces you type; one is good as 100.

Note also that one or more blank lines denote the end of a paragraph.

圖 5.1: 控制空白與段落

距相同。其次，第 1 行尾巴的 `sentences` 和第 2 行開頭的 `are` 之間並未留有空白，但是排版結果兩個單字之間也有正確的間隔，原因是 `sentences` 之後換新行。在顯示器上雖然第 1 行末端看不到任何符號，但電腦檔中該處隱藏一個換行指令。排版時，碰到此換行指令，`TeX` 就當作一空白處理。

本例子的兩段文字之間有一空行，其作用是確認以上段落結束，以下為新起一段之開頭。如果不留空行，我們必須在第一段之末加上 `\par` 指令。指令 `\par` 為英文 `paragraph` 之縮寫，意義為段落。和空白鍵的情形類似，空一百行和空一行的作用完全一樣，都是表示要另起新的段落。如果沒有另外設定，新段落開始的一行，行首會自動內縮一點。英文排版中，這稱為 `indent`。

排版之後，英文單字之間留有適當間隔，標點符號之後也留有空白。某些標點符號，如句點或問號，代表一段話的結束，讀者的眼睛在此地方應該有小停頓。空白多一點表示停頓的時間長一些。因此，句點之後的空白應該比逗點之後的大一些，因為前者表示整句話的結束，後者只是一長句子中的小休。為了使 `TeX` 能正確處理標點符號之後的間隔，輸入文稿時標點符號之後應留一空白，或者換新行。

為了提高可讀性，`TeX` 在英文句點 `.`，問號 `?`，冒號 `:`，及驚歎號 `!` 之後所留的空白，比單字之間的空白或者逗點之後的空白都要大一些。但是，有時候英文句點並不代表句子結束。舉一個簡單的例子，如果你輸入 `Mr. know-all`，排版之後將如第 1 行所示。

Mr. know-all
Mr. know-all

Mr. know-all
Mr.\ know-all

Mr. 單字的小點並不是英文句點，但 `TeX` 誤把它當作句點，因此 `Mr.` 與 `know-all`

之間的空白拉大了一些。若句點之後不特別加多空白，排版結果應該是第二行所示。有人會認為這一點差異沒有那麼了不得。不過，在專家的眼中這個差別是很重要的。要使句點之後的空白大小正確，我們應使用 `_` 指令，亦即，反斜線之後加上一空白。因此，正確的輸入方法是第 2 行所示：`Mr._know-all`。

英文句點之後若緊接著右圓括號或者英文引號，`ETEX` 也會將之解釋成是句子結束而加大空白。但是，偶而會有例外情況，如：

Famous coffee beans (Santos, etc.) are expensive.	Famous coffee beans (Santos, etc.)\ are expensive.
------------------------------------------------------	-------------------------------------------------------

輸入時，右圓括號之後加上強制空白指令，以免空白太大。

5.3 中文稿件輸入原則

輸入中文稿時，文字之間是否留空白，以及在何處更換新行，須特別小心。首先，輸入時中文字之間不應該留有空白。如果中文字之間插入空白鍵，排版之後兩字之間距會變大。底下是一個例子：

輸入中文文 稿時，我們對於 空白 及換新行要很小心。	輸入中文文 稿時，我們對於 空白及換新行要很小心。
-------------------------------	------------------------------

因為「文 稿」兩個字之間有一空隔，排版之後兩字之間距並不正確。此外，上一節曾說明更換新行和留空白的的作用相同。本例中「對於」兩字與下一行開頭的「空白」之間因為更換新行，排版之後字距也加大了。

那麼中文稿應該在那裡更換新行呢？如前所述，標點符號之後本來就要留一空白，以使排版結果美觀。因此，中文稿也應該在標點之後更換新行。綜合以上所述，輸入中文稿請謹記下列兩項原則：

- 句子內的中文字之間請勿留下空白；
- 換新行請盡量在標點符號之後。

根據這兩項原則，上面句子中「文稿」兩個字中間不應留空白；「空白」兩個字之後不應該換行；換行應該在逗點之後。

不過, 如果你的中文輸入已養成習慣, 在固定長度就按 [Enter] 鍵換行, 則解決問題的方法是在執行 `cwTeX` 時加入 `-c` 選項。譬如, 若文稿檔名為 `test.ctx`, 轉換中文字時應執行下列指令:

```
c:\xtemp>cwTeX -c test
```

加入選項之後, 若遇有兩個中文字間插入換行指令之情況, `cwTeX` 會將換行指令刪除。若是使用 `WinEdt` 軟體, 我們可將以上之設定加入 [F9] 功能鍵內, 以方便使用。設定方法如下, 開啓 `WinEdt` 視窗上方之:

Option | Menu Setup | &Accessories | cwTeX

將 `Utility` 空格內之 `cwTeX.exe` 改為 `cwTeX.exe -c` 即可。

有時候, 我們會碰到非常長的句子, 若要等到標點之後再換行, 輸入時不很方便。此時, 我們可以使用註銷指令 `%`。如前所述, 一行文字中若出現註銷指令, 該指令及其後所有的文字符號都不處理。譬如,

很長的英文單字, 例如 `percholoe-
thylene`, 可以用註銷指令切成兩行;
中文亦然。

很長的英文單字, 例如 `percho%
loethylene`, 可以用註銷指令切成%
兩行; 中文亦然。

因為第 1 行之末有 `%` 指令, 這等於是第 2 行開頭的 `loethylene` 直接緊接在第 1 行尾巴的 `percho` 之後, 因此可排版出正確結果。同理, 「切成」兩字之後緊接著加上 `%` 指令, 因此排版結果不會出現多餘的空白。

5.3.1 標點符號

`TeX` 所定義的標點符號共有下列 16 個:

, . : ; ? ! ‘ ’ () [] - / * @

在 `cwTeX` 系統中, 以上 16 個標點符號用於英文, 也用於中文內。但中文另外有 11 個標點符號是英文沒有的, 例如中文句點。中文輸入法都有特定的方法以鍵入中文標點符號。為了方便起見, 我們在 `WinEdt` 軟體內設定鍵入中文標點符號之功能鍵。請見表 5.4。例如, 在 `WinEdt` 視窗內按 `Ctrl+Shift+o` 即出現中文句點。

表 5.4: 輸入中文標點符號之功能鍵

標點符號	WinEdt 功能鍵	標點符號	WinEdt 功能鍵
、	Ctrl+Shift+i	「	Ctrl+Shift+h
。 。	Ctrl+Shift+o	」	Ctrl+Shift+j
?	Ctrl+Shift+l	『	Ctrl+Shift+f
,	Ctrl+Shift+k	』	Ctrl+Shift+g
;	Ctrl+Shift+m	《	Ctrl+Shift+r
。	Ctrl+Shift+d	》	Ctrl+Shift+t
<	Ctrl+Shift+y)	Ctrl+Shift+u

Ctrl+Shift+k 按鍵可產生英文逗點, Ctrl+Shift+m 按鍵為分號。另外, 中文數字「。」之內碼為 A2AF。在倉頡輸入法下, 鍵入全型的 0 亦可得到結果。

目前各式各樣的中文輸入方法中, 有不少是採所謂的「全型輸入」方式輸入標點符號。譬如, 中文句點在顯示器上占用相當於兩個英文字母的位置。這是所謂的「全型」。以上所列出的中文標點符號, 全部都必須以全型方式輸入。相對的, 中英文共用之逗點或者問號, 可能採全型輸入, 也可能是採半型輸入。所謂「半型」是指直接英數狀態下的英文逗號, 因此它所占用的空間和英文字母或阿拉伯數字相同。

使用全型輸入中文標點符號時, 雖然顯示器上看起來該標點之後有一空白, 其實不然。你必須再鍵入一空白才算數。輸入全型標點符號時若未留空白, 排版結果中文字與標點符號之間距可能仍然是正確的。原因是 `cwTeX` 中文轉換程式會辨識常見的錯誤, 並作適當的修正。雖然 `cwTeX` 有此功能, 使用者最好還是養成正確的輸入習慣。

`TeX` 會在英文句點, 問號, 冒號及驚歎號之後多加一點空白, 以提高可讀性。同樣的, `cwTeX` 也在中文句點之後多加上一點空白。但是, 有些標點符號之前或之後是不須留空白的。譬如:

小明說:「公主大叫『救命啊!』。」

小明說:「公主大叫『救命啊!』。」

本例中, 驚嘆號之後的三個標點符號之間, 不應加上任何空白。如果插入空白, 結果反而不佳。

5.3.2 英文引號與破折號

英文標點符號用於中文稿中唯一要注意的是引號。如果你要排版‘測試’或‘test’，輸入之指令分別是：‘測試’與‘test’。請特別注意，左單引號和右單引號是不同的。個人電腦中，左單引號字鍵通常置於鍵盤左上角；右單引號鍵則置於鍵盤的右下角。

如果要排版雙引號，譬如，“測試”，應如何輸入呢？一般鍵盤的右下方有一個雙引號的字鍵，因此很多人會直接輸入“測試”。但是，排版結果卻變成：”測試”。正確的輸入是：“‘測試’”。亦即，左雙引號是連續輸入兩個左單引號；右雙引號則是連續輸入兩個右單引號。左右引號未分辨清楚，是初學者常犯的錯誤，應特別小心。

如果你必須同時使用單引號與雙引號呢？譬如，你碰到下面的句子：

“‘Five’ or ‘Hive?’” she asked.

若輸入“‘‘Five’ 我們不曉得到底是前兩個單引號構成雙引號，或者是後面兩個。解決此一困擾的方法是使用 \, 指令，在兩個引號之間加入一小小的空白。因此，正確的輸入方法如下：

“\, ‘Five’ or ‘Hive?’” she asked.

英文使用的破折號由短到長共有三種。最短的稱為 hyphen 是用於連接英文單字，如 mother-in-law。次長的破折號稱為 en-dash，通常用於連接數目字。譬如，頁 34–56，或者，1945–1950 年。要產生 en-dash，你必須接連著輸入兩個 hyphen: --。最長的破折號是由三個 hyphen 組成，其用法相當接近中文之破折號：台電公司即將限電 — 如果天氣不轉涼的話。

以上的分類適用於 Knuth 所設計的 Computer Modern 字體。如果我們使用 PostScript Type 1 字體排版，其區分並無如此細緻。這些字體通常並未區分 hyphen 與 en-dash。本書正文使用的是 Minion 字體，排版頁碼範圍時若直接使用 hyphen 指令，結果為：頁 12-35；若使用 en-dash，結果為：頁 12–35。

還有一個和破折號樣子很像的符號是減號，不同的是其筆劃較粗。要排版減號，應該輸入：\$-\$。前後加上兩個錢號 \$ 的作用是使 L^AT_EX 進入數學模式。譬如，

表 5.5: 破折號與引號

名稱	指令	例子
hyphen (-)	-	motor-car, mother-in-law
en-dash (–)	--	頁 12–35, 1981–1990 年
em-dash (—)	---	I saw them — three of them
減號 (−)	\$-\$	5 − 2 = 3

輸入 `-100`, 排版結果為: `−100`, 若不進入數學模式, 排版結果為 `-100`。以上所述標點符號的用法歸納於表 5.5; 數學式排版將於第 9 章說明。

有些人以連續三小點 (甚或六小點) 代表一段省略的話。在 \LaTeX 中, 你可以用 `\dots` 指令產生三小點。例如指令 `\dots\dots` 產生: `.....`。最後, 謹列舉輸入文稿之原則, 以供參考。

- 逗號或句號不宜與前接文字分開; 因此, 「鍵入雙引號,」為錯誤, 因為逗號之前不應留空白。正確的排版是: 「鍵入雙引號,」。相反的, 逗號或句號之後應留空白。因此, 「譬如, 下面的句子 ...」為錯誤, 正確應該是「譬如, 下面的句子 ...」。
- 同理, 段落前端之標點符號也不宜和後接文字分開。譬如, 「昭和 20 年 (1945 年) 日本投降」比「昭和 20 年 (1945 年) 日本投降」為佳, 阿拉伯數字 “1945” 之前應緊接圓括號, 「年」之後也不應留空白。另外, 「昭和 20 年 (1945 年) 日本投降」也不佳, 左圓括號前面與右圓括號後面應留出空白。
- 阿拉伯數字前後有無空白間隔皆可, 但以不留空白為佳。譬如, 「西元 1945-1950 年間的台灣」, 輸入時中文與阿拉伯數字間並未留空白; 但 `cwtext` 會自動在阿拉伯數字與中文字之間加入一點小空白。若自行留空白, 結果變成「西元 1945-1950 年間的台灣」。
- 英文字夾於中文句子中, 前後則以留出空白為宜。譬如, 「最短的稱為 hyphen 是用於」比「最短的稱為 hyphen 是用於」易讀。

以上之細節看似瑣碎, 但對於排版結果大有影響。欲排版高品質文稿, 平常請多用點心, 觀察、欣賞好的排版作品, 學習其品味與風格。專業排版對於標點符號之規範, 請見 [Bringhurst \(1996\)](#)。以上的說明是針對一般文稿而言, 有關於數學文稿之輸入原則, 請見 9.2.1 節 (頁 155)。

5.4 排版其他西方國家文字

除了英文之外, \LaTeX 也可以排版德文、法文等歐洲國家文字。網路上可以找到幾套專門用來排版某國文字之巨集套件。(有關於巨集套件的意義及使用方法, 請見第 7 章開頭之說明。) 譬如, `german.sty` 可用以排版德文, `french.sty` 可用來排版法文。不過, 由這些巨集套件所衍生的 `babel` 巨集套件是使用最方便, 彈性最大的系統, 作者是 Johannes Braams。

`babel` 系統可處理二十多種歐洲國家文字, 對每一個國家的文字都提供下列功能:

- 將英文標題字替換為該國文字之標題,
- 提供指令以供輸入特別字母,
- 提供正確的音節斷音 (hyphenation)。

除此之外, `babel` 系統允許在同一篇文稿內排版兩種以上的文字。舉一個例子來說, 如果某篇文稿內同時有英文與德文, 排版時在全文設定區輸入下列指令:

```
\usepackage[english,german]{babel}
```

方括號內選定排版之文字, 最後出現的 `german` 表示文稿內容以德文為主。

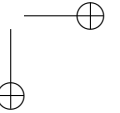
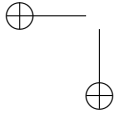
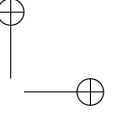
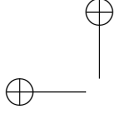
文稿一開始排版幾節德文段落之後, 若接下來出現英文, 應在英文段落之前鍵入下列一行指令:

```
\language=1
```

則英文段落內之音節斷字即能正確處理。再接下來若遇有德文, 則使用

```
\language=0
```

指令即可回到德文模式。有關於 `babel` 系統的使用細節, 請參考巨集檔案內附之說明檔, 或者參考 [Goossens et al. \(1994\)](#), 9.2 節之說明。



6 選用字體

開始著手排版一本書或一篇文稿之前，應先想好版面之整體設計，譬如，版面尺寸、正文字體、章節標題格式、註解的字級 (font size) 等等。使用 \LaTeX 的好處是我們可以選用現成的版面格式，不須花太多時間在版面設計上。舉例來說，如果正文使用 12pt 字體，則文稿中註解之字體會自動選用 10pt，行距也會小一些。又如文稿中若區分章、節，其標題會自動選用較大的字體。

\LaTeX 所設計的版面已考慮了專業排版的規範。除非你有專業排版的知識，否則使用現成的格式排版，其結果通常優於自行設計者。使用現成的格式，一方面節省時間，另一方面免於犯一些常見的排版錯誤，這是其優點。但是，也有人認為 \LaTeX 所排版之書籍其版面格式都很接近，不易顯出個人風格。要變更內定之版面並不難；特別是在新版 \LaTeX 中，使用者有相當大的調整空間。

6.1 字型規格

介紹字體指令之前，我們先簡單說明字體設計的概念。所謂字體 (font)，簡單來說是指文字的某一種書寫方法。在排版中，字體選擇占著舉足輕重的地位。Knuth 教授當初發展 \TeX 排版系統時，同時又發展一套設計字體的軟體 METAFONT。原始 \TeX 與 \LaTeX 系統所使用之英文字體全部都是利用 METAFONT 軟體設計出來的。這些字體之字型檔名大部分是以 `cm` 開頭，代表 Computer Modern 字體。譬如，羅馬字體 10pt 之字型檔就稱為 `cmr10`。

傳統的鉛字排版中，字型直接刻在鉛塊上。電腦排版發展之後，每一種字體內各單字或字母之形狀則儲存為電腦檔案。過去一、二十年中，字體設計之技術有相當大的進步。其中，美國 Adobe 公司所發展之 Type 1 字體 (又稱為 PostScript 字體) 是最重要者。Type 1 字體是所謂的描邊字體 (outline font)，每一個單字或字母之形狀都以數學式描繪出來，列印時再填為實心字。描邊字體的好處是可以隨意放大，而不會嚴重失真。自 1980 年代中期以來，Type 1 字體日益普及，幾乎

變成是電腦排版的字體標準。因為如此,新版 \TeX 一個重要的發展方向就是讓排版者可以取用 Type 1 字體。我們將在第 13 章介紹 Type 1 字體的使用。

中文電腦字型之技術近幾年來也有很大的進步,特別是在 Windows 系統開始推廣之後。Windows 主要使用另外一種字型規格,稱為 True Type,但也可以使用 Type 1 字型。中文電腦字型的設計面臨許多困難,最主要的是中文字數實在太多。在英文中,一套字體通常不超過 256 個字母,其中包括大小寫字母、數字、標點符號等等。但是,一套中文明體字可能超過一萬字。以 Big-5 內碼為例,常用字有 5,401 字,次常用字則有 7,652 字。因此,創造一套中文字體所耗人力物力相當驚人。這是為什麼英文字體的選擇很多,而中文字體的選擇卻很有限。

在 \TeX 中,我們可以選用不同的英數字體,也可以將字體放大或縮小。不過, \TeX 之字體指令只能處理英文與數字字體,對於中文字體並無作用。中文字體的變換與放大縮小必須靠 c\TeX 的中文字體指令。以下兩節進一步說明之。

6.2 選用英數字體

Knuth 一共造出七十多種字體。除了英文與數字之外,還有不少特殊符號是爲了排版數學用的。排版時,如果你不指定字體, \TeX 就選用 10 點字級 (font size) 之羅馬字體編排, c\TeX 內定的中文字體則爲 10 點明體字。

在舊版的 \TeX 系統中,選用字體的指令彈性較小。1990 年代初期,兩位德國專家 Frank Mittelbach 與 Rainer Schöpf 設計了一套「新式字體選用法」(new font selection scheme, 簡稱為 NFSS)。此套方法出版之後,頗受歡迎,現在已經變成新版 \TeX 系統之一部分。本節以下所介紹的是新版之字體指令。舊版的 c\TeX 中,英文字體指令對中文字無效。但從本版開始, \TeX 改變字體大小之指令對中文也有效。

英文字體式樣甚多,但依 NFSS 分類,字體可以三種特徵區分:

- 字形 (font shape),
- 字體序列 (font series),
- 字體族 (font family)。

字形又進一步區分爲:直形 (upright), 意大利斜體 (*italic*), 數學斜體 (*slant*), 與 SMALL CAPITAL 等四類。字體序列則包括中體序列 (medium series) 與粗體序列

表 6.1: 英文字體指令

字體分類	標準字體指令	宣告字體指令	簡化宣告指令
字形 (<i>shape</i>)			
Upright	<code>\textup{ text}</code>	<code>{\upshape text}</code>	
<i>Italic</i>	<code>\textit{ text}</code>	<code>{\itshape text}</code>	<code>{\it text}</code>
<i>Slant</i>	<code>\textsl{ text}</code>	<code>{\slshape text}</code>	<code>{\sl text}</code>
SMALL CAPS	<code>\textsc{ text}</code>	<code>{\scshape text}</code>	<code>{\sc text}</code>
字體序列 (<i>series</i>)			
Medium	<code>\textmd{ text}</code>	<code>{\mdseries text}</code>	
Boldface	<code>\textbf{ text}</code>	<code>{\bfseries text}</code>	<code>{\bf text}</code>
字體族 (<i>family</i>)			
Roman	<code>\textrm{ text}</code>	<code>{\rmfamily text}</code>	<code>{\rm text}</code>
sans serif	<code>\textsf{ text}</code>	<code>{\sffamily text}</code>	<code>{\sf text}</code>
Typewriter	<code>\texttt{ text}</code>	<code>{\ttfamily text}</code>	<code>{\tt text}</code>

(**bold series**)。字體族則包括以下三類: 羅馬字族 (Roman family), sans serif (無裝飾邊) 與打字機 (typewriter) 字族。以上三種分類中, 字體族的範圍最大。譬如, 在羅馬字族之下可區分為中體序列與粗體序列, 而中體序列之下可進一步區分為直形、意大利斜體形等; 請見表 6.1。

文稿中任何地方都可以以下指令改變英數字體。字體指令有兩種格式, 第一種指令格式與一般的指令類似, 亦即將欲改變字體之段落置於大括號內, 我們將稱之為「標準字體指令」。譬如, 英文 `text` 要改變為粗體字 **text**, 標準字體指令為 `\textbf{text}`; 要變更為意大利斜體 *text*, 指令為 `\textit{text}`。如果是要改變為數學斜體字 *text*, 指令為 `\textsl{text}`。請注意, 數學斜體與意大利斜體並不相同, 前者主要用於數學式子內。

除了標準字體指令之外, 我們也可使用「宣告字體指令」(declaration command) 變更英文字體。事實上, 這是舊版 L^AT_EX 改變字體的標準方法。宣告字體指令提供三種格式, 以下以粗黑體為例說明之。

- 第一種格式是在文稿中任何地方直接下 `\bfseries` 指令, 從該點開始所有的英數文字都變成粗體字。
- 第二種下指令的方式與第一種指令相同, 但指令 `\bfseries` 簡化為 `\bf`。大部分宣告字體指令都可以使用簡化指令。譬如, `\itshape` 指令可以簡化為

`\it`, 而 `\rmfamily` 指令可簡化為 `\rm`。

- 第三種方法是以指令環境之方法改變字體, 譬如:

```
\begin{bfseries} text \end{bfseries}
```

指令環境內所有英數文字都變成粗體字。

以上三種指令格式中, 第二種格式最為簡單。

上面所介紹的標準字體指令格式是以大括號界定字體變更的範圍, 宣告字體指令也可以大括號界定範圍, 但是, 大括號必須將字體指令本身也涵蓋在內。例如, 以下指令將大括號內三個英文字改用數學斜體編排:

```
{\sl switch to slant}
```

右大括號之後又回復原先的英數字體。底下是一個較複雜的例子, 但字體指令之運作原理不難理解。

*Started with italic, switch to Roman, switch to slant, back to Roman, make one word **bold**, simulate typewriter .*

```
\it Started with italic,
\rm switch to Roman, {\sl switch
to slant}, back to Roman,
make one word \textbf{bold},
simulate {\tt typewriter}.
```

本例是以 `\it` 指令選用斜體字形; 但我們也可以用 `\em` 替代 `\it`, 效果相同。

本例子內之字體指令主要是變更字體序列與字形。如果硬碟內安裝了他種 PostScript 字型檔, 如 Times Roman 或 Garamond, 我們也可以在文稿內選用另一字體族。安裝 PostScript 字型檔的方法, 請見第 13 章; 選用字體之指令, 請見 13.4.2 節 (頁 271)。

一般的 \LaTeX 宣告指令, 包括字體指令在內, 如果置於大括號內, 指令之效力僅限於大括號內。 \LaTeX 提供許多指令環境, 例如排版表格的 `tabular` 指令環境。宣告指令如果置於特定指令環境內, 指令環境結束之後, 宣告指令之效力也消失。如果希望指令之效力僅限於某一範圍, 最簡單的方法是以左右大括號界定其範圍。

在英文打字稿中常以加底線的方式強調文句。在 \LaTeX 中, 加上底線的指令是 `\underline`。例如:

表 6.2: 字體級數對照表

字體級數	放大倍數	字體點數
0	$1.2^0 = 1.0$	10
h	$1.2^{0.5} \cong 1.095$	10.95
1	$1.2^1 = 1.2$	12
2	$1.2^2 = 1.44$	14.4
3	$1.2^3 = 1.728$	17.28
4	$1.2^4 \cong 2.074$	20.74
5	$1.2^5 \cong 2.488$	24.88

說明: 字體級數 h (half), 代表半級。因此, 對應之放大倍數為 $1.2^{0.5} \cong 1.095$ 。

此句話加上底線。

`\underline{ 此句話加上底線。}`

底線指令可用於中文, 亦可使用於英文。不過, 正式排版中加底線的作法較少見, 通常是以改變字體的方式為之。

6.2.1 字體相對大小指令

除了更換字體之外, 我們也可以改變字級。前面已說明, 大部分字體之設計尺寸為 10 點。不過, 我們可以將 10 點之字體放大成 14 點, 或者縮小為 8 點。所謂 14 點的字體, 其橫寬約為 10 點字體的 1.4 倍; 相對的, 8 點字體的橫寬為 0.8 倍。依原先設計, 大部分字體都具有下列的點數 (由小至大): 5, 6, 7, 8, 9, 10, 10.95, 12, 14.4, 17.28, 20.74 與 24.88。

你或許會覺得奇怪, 為什麼字體點數為 10, 10.95, 12, 14.4, ..., 而卻沒有 13 點或 16 點? \TeX 字體在放大或縮小時, 其倍數是以 1.2 的次方為單位, 而次方數為 0, 0.5, 1, 2, ...。若放大次方數為 1, 字體點數為 $10 \times 1.2^1 = 12$ 。因此, 12 點之字體也可以說是放大 1 級之字體。若放大次方數為 2, 則字體點數為 14.4, 簡稱為 14 點。因此, 當我們說 14 點 (或 14 級) 字體時, 實際上它是 14.4 點之字體。同樣的, 若放大次方數為 4, 則字體點數為 $10 \times 1.2^4 \cong 20.74$, 簡稱為 20 點字體。因此, 20.74 點之字體也可以說是放大 4 級。表 6.2 列出字體級數對照表。

選擇英文字級有兩種方法，一種是直接選定字級；另一種則是以相對大小指令選擇字級。我們首先介紹相對大小之指令。字級相對放大之指令如下：

`\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`

以上指令所產生之字體放大之效果如下所示：

normal, large Large LARGE huge Huge

其中，`\Large` 字級比 `\large` 大一些；`\LARGE` 又比 `\Large` 大一些；餘此類推。相反的，字級相對縮小指令依序為：`\small`，`\footnotesize`，`\scriptsize`，`\tiny`。其中，`\tiny` 所產生的字體最小。

normal, small footnotesize scriptsize tiny

相對大小指令所產生的字體，其實際尺寸上並不是固定的，而是相對於正文之字級而定。正文字體之字級稱為 `\normalsize`。如果我們選定正文字級為 10 點，`\normalsize` 意指 10 點之羅馬字體。此時，`\large` 指令將選用 10.95 點之字體。但是，若正文字級為 12 點，則 `\large` 指令將選用 14.4 點之字體。正文字體之選定由 `\documentclass` 指令設定，例如以下指令選用 12 點字體與短文格式：

`\documentclass[12pt]{article}`

如果文稿某處選用了較大字體，則 `\normalsize` 指令可以變回內定的正文字級。

6.2.2 選擇字級與行距

選用較大字體時，須同時選用較大的行距，版面才會美觀。另外，中文字之筆劃較英文字複雜。因此，英文稿之行距可以小一些，但中文稿之行距必須加大。如果中文稿中夾有一段英文，其行距必須仔細調整才會美觀。

「行距」指的是本行到下一行的距離。但是，從本行的那一點到下一行的那一點呢？版面的每行文字都有一條所謂的「基線」(baseline)，行距即指兩條基線的距離。下例中的兩條橫線即為基線，我們設定之行距為 22 點，也就是這兩條細橫線的垂直距離。

— An example of baseline: 基線。 —
 — Another baseline. —

由此例子亦可看出來，英文小寫字母恰坐落在基線上，而中文字的底部則比基線還要低一些。

行距可以由 `\baselineskip` 定義之，但每一種字級各有內定之行距。當我們使用相對放大指令時，行距也隨之而變更。因此，在英文稿中，我們幾乎不須擔心行距調整的問題。但是，有些時候我們需要改變行距。譬如，文章初稿之行距通常會大一些以利修改。又如，中文字之筆劃複雜，因此中文稿之行距通常也大一些，以利閱讀。

欲更改全文之行距，最簡單的方法是在文章一開頭即加以設定。譬如，要將全文之行距放大為內定值之 1.15 倍，可使用以下的指令：

```
\renewcommand{\baselinestretch}{1.15}
```

此一指令通常是置於全文設定區，也就是在 `\begin{document}` 指令之前。另一個功能相同之指令是 `\linespread`。在全文設定區鍵入下列指令也可以將行距加大為 1.15 倍：

```
\linespread{1.15}
```

請注意，以上指令將把全文各部分之行距通通放大為原來的 1.15 倍。文稿中若有排版註解，其行距也會同比率加大。若要進一步微調註解之行距，請見 8.6 節（頁 143）之說明。反之，如果全文各部分之行距全部要改為某固定值，應使用下列指令：

```
\setlength{\baselineskip}{15pt}
```

以上指令將全文行距一律變更為 15pt。

上述兩項指令也可以出現在正文中，但是其發生效果的時刻不同。在同一段落中若變更 `\baselineskip` 數次，只有最後一次之變更值有效。而且，每當字級改變時，行距即變回內定值。相反的，若 `\baselinestretch` 指令是下於行文當中，則只有在下一次字級變更時，它才發生效果。如果希望它即時產生效果，我們可以在指令之後加上 `\small\normalsize`，其功能是把字體縮小一級之後，又立刻回復原來大小。因為字級已變更，行距指令即可發揮作用。

另外一種改變字級與行距的方法是使用 NFSS 之 `\fontsize` 指令。假設要選用 14.4 點字體，同時行距要變成 20 點，指令為：

表 6.3: 選用字體與行距之巨集指令: mymacro.tex

```

\newcount\fs
\def\sz#1#2{\fs=#1#2
\ifnum\fs=10\fontsize{10}{12.5pt plus.2pt minus .1pt}\selectfont
\else\ifnum\fs=11\fontsize{10.95}{17pt plus.3pt minus.2pt}\selectfont
\else\ifnum\fs=12\fontsize{12}{18.5pt plus.3pt minus.2pt}\selectfont
\else\ifnum\fs=14\fontsize{14.4}{20pt plus.4pt minus.3pt}\selectfont
\else\ifnum\fs=17\fontsize{17.28}{22pt plus.4pt minus.3pt}\selectfont
\else\ifnum\fs=20\fontsize{20.73}{28pt plus.5pt minus.3pt}\selectfont
\else\ifnum\fs=25\fontsize{24.88}{33pt plus.5pt minus.3pt}\selectfont
\fi\fi\fi\fi\fi\fi}

```

```
\fontsize{14.4}{20pt plus.5pt minus.4pt}\selectfont
```

第一個大括號中之 14.4 是選用 14.4 點字體；第二個大括號中之數字選定行距為 20 點，伸縮彈性為正 0.5 點與負 0.4 點。伸縮彈性部分之指令可有可無，但最好是有，因為 \TeX 較容易編排版面。不過，行距之伸縮彈性也不宜太大，否則每一頁版面之行距可能大小不一，版面看起來的感覺並不好。

利用 `\fontsize` 指令可以同時改變字級與行距。如果經常須變動字級與行距，使用表 6.3 之巨集指令較方便。此巨集指令名為 `mymacro.tex`，已置於 $\text{cw}\TeX$ 系統內。欲使用此巨集指令，請在全文設定區加入下列一行指令：

```
\input mymacro
```

若要選用 17.28 點之字體（行距設定為 22pt），只要鍵入 `\sz17` 即可；如果要選用 12 點之字體（行距為 18.5pt），則鍵入 `\sz12`；餘此類推。`mymacro` 巨集指令之例子中設定 12 點字體之標準行距為 18.5 點，另再加上伸縮彈性。不過，行距及伸縮彈性都可以自行修改。

有時候我們使用 `\fontsize` 指令的目的不在改變字級，而只是要變更行距。圖 6.1 的例子裡，我們以 `\sz11` 指令選用 11pt 之字體。你可能注意到，本例中的英文斜體字指令 `\textsl` 同時也將中文明體字轉換為斜體；英文粗體字指令 `\textbf` 則同時將中文字轉換為粗體字。但是，上面曾說過，英文字體指令僅能轉換英文字，對中文字無效。為什麼中文字體也同時改變呢？

為了方便選用中文字體， $\text{cw}\TeX$ 系統提供簡單的巨集指令功能。本例中，我

以 11pt Roman 字體排版, 行距為 17.6pt。switch to *italic type*, 選用斜體 *slant type*, switch back to Roman and make one word **bold** 粗體字。

```
\input mymacro
\ctxfdef{\textsl}{\ctxfms}
\ctxfdef{\textbf}{\ctxfbb}
...
\sz11
以 11pt Roman 字體排版, 行距為 17.6pt。
\em switch to italic type,
\textsl{ 選用斜體 slant type},
\rm switch back to Roman and
make one word \textbf{bold} 粗體字。
```

圖 6.1: 變更字體

們在全文設定區加入兩行指令:

```
\ctxfdef{\textsl}{\ctxfms}
\ctxfdef{\textbf}{\ctxfbb}
```

其功能是指定將 `\textsl` 指令下之中文字以明斜體 `\ctxfms` 排版; `\textbf` 指令下之中文字則以粗黑體 `\ctxfbb` 排版。請見 6.3 節進一步的說明。

6.2.3 選用任意點數之英數字體

依原先設計, \TeX 僅能使用特定尺寸之英數字體。不過, Knuth 所設計之 Computer Modern 字體已經被轉換成 Type 1 描邊字型格式。若使用新格式之字型排版, 我們即可選用任意點數之字體。其方法是在全文設定區引用 David Carlisle 之 `type1cm` 巨集套件:

```
\usepackage{type1cm}
```

引入此巨集套件之後, 英數字級可以透過 `\fontsize` 指令任意選擇。譬如, 下列指令

```
\fontsize{13.5}{20pt}\selectfont
```

即選用 13.5pt 之英文/數字字體。

6.3 選用中文字體

上一節所介紹之字體指令只能變換英文及數字之字體, 對於中文字並無作用。中

<code>\ctxfml</code>	石創際手祝迴	<code>\ctxfrl</code>	石創際手祝迴
<code>\ctxfrm</code>	石創際手祝迴	<code>\ctxfrr</code>	石創際手祝迴
<code>\ctxfmb</code>	石創際手祝迴	<code>\ctxfrrb</code>	石創際手祝迴
<code>\ctxfmu</code>	石創際手祝迴	<code>\ctxfrru</code>	石創際手祝迴
<code>\ctxfmx</code>	石創際手祝迴	<code>\ctxfrrx</code>	石創際手祝迴
<code>\ctxfbl</code>	石創際手祝迴	<code>\ctxfrrl</code>	石創際手祝迴
<code>\ctxfb</code>	石創際手祝迴	<code>\ctxfrrf</code>	石創際手祝迴
<code>\ctxfbb</code>	石創際手祝迴	<code>\ctxfrrk</code>	石創際手祝迴
<code>\ctxfbu</code>	石創際手祝迴	<code>\ctxfrrk</code>	石創際手祝迴
<code>\ctxfbx</code>	石創際手祝迴	<code>\ctxfrrkx</code>	石創際手祝迴
<code>\ctxfll</code>	石創際手祝迴		
<code>\ctxfllb</code>	石創際手祝迴		

圖 6.2: 中文字體範例

文字體的變換必須利用 `cwTeX` 之字體指令。在 `cwTeX` 第2版以前, 中文字體指令必須同時選用字體與大小, 例如, `\m12` 選用中文明體 12 點字體。2.1 版開始, 透過 `ETEX` 之新式字體選用法 (NFSS), 使用者可以僅選定字體種類, 字體大小則由 `ETEX` 控制。這使得中文字體指令更有彈性, 指令也更為簡化。更重要的是, 中英文字體之大小相同, 不會有大小不一的情況出現。

6.3.1 新式字體選用法

欲使用中文之新式字體選用法, 你必須使用 `cwTeX.exe` 13.6 或更新版本。字體指令很簡單, 中明體指令為 `\ctxfrm`, 粗明體指令為 `\ctxfmb`, 粗黑體指令為 `\ctxfbb`。圖 6.2 以 14 點字級為例, 列示 `cwTeX` 現有的中文字體及引用字體之指令。

以上之指令僅選用字體, 大小則由 `ETEX` 控制。在新式字體選用法之下, 如果全文內未下任何中文字體指令, 則 `cwTeX` 將自動選用明體字。舉例言之, 第 2 章第 1 個例子中, `\documentclass[12pt]{article}` 文件類別指令選用 12pt 字體大小。因為加入 12pt 選項, 若全文未加入任何中文字體指令, 排版時 `ETEX` 將自動選用 12 點明體字型。又如, 文稿內若使用 `\footnote` 指令排版註解, 其內之中文字將自動縮小為 10 點大小。

文稿內欲改變中文字體, 指令如下:

... 總督府推動{\ctxfbb 資本主義化}的政策, ...

即可將「資本主義化」改為粗黑體。中文字體之選擇是以 \ctxfm, \ctxfbb 等指令為之。如果只是要改變中文字大小, 應直接使用 \TeX 指令。我們可使用相對字體大小指令, 如 \large, \small 等; 或者以 \fontsize 指令直接選用特定點數, 請見上一節之說明。

圖 6.2 所列每一種字體尚可作水平縮小或傾斜之變形。例如, \ctxfbbs 指令選用粗黑斜體字, \ctxfbbe 為粗黑狹長體字, 而 \ctxfbbes 則選用粗黑狹長斜體字, 同理, 選用明狹長體之指令為 \ctxfme; 楷斜體之指令為 \ctxfks。中文變形字之式樣, 如傾斜角度與水平縮小比例, 皆可以自行設定。中文變形字之例子及相關討論, 請見圖 13.1 (頁 260)。

以上所介紹的字體是用於橫排之文稿。除了橫排字體外, cwTeX 尚提供直排字體。譬如, \ctxfvm 選用直排明體字; \ctxfvbb 則選用粗黑體。換言之, 橫排字體指令前端加上英文字 v, 即選用直排字。直排例子請見第 2 章例 7。直排字體是用於直排文稿, 而直排文稿之頁面設計, 如頁眉與頁足等, 與橫排文稿不同。直排文稿之種種處理細節尚有待進一步測試。

cwTeX 中文字體指令之有效範圍與 \TeX 英文字體宣告指令相同, 若指令下於大括號或指令環境內, 該指令僅於該範圍內有效。譬如, 以下三行例子中都在文字當中改變字體。因為中文字體指令都是包圍在大括號或指令環境內, 故字體指令僅於該範圍內有效。

```
... \footnote{...} ...
... \author{... \ctxfk ...} ...
... \begin{tabular} \ctxfbb ...\end{tabular} ...
```

第 2 行例子以 \author 指令排版作者名字。在 \ctxfk 指令之前, 以原先之明體字排版, 之後則改用楷體。但是, 在右大括號之後又回復原先之明體字。

第 2 章例 2 排版文稿題目的指令如下:

```
\title{ 台灣長期總產出之變動}
\author{ 吳聰敏\thanks{
  作者任教於台大經濟系。}}
```

文稿題目之中文字體並未作任何設定, 故以內定之明體排版。作者名字也未加入

任何字體指令, 因此, 也是以明體字編排, 但 \LaTeX 會自動加大字體大小。作者名字之後的 \backslashthanks 指令是用以排版謝詞, 其字體仍然是明體字, 字體尺寸則比正文略小。

行文中改變字體一定要加上左右大括號, 以確定字體變更的範圍。有時候, 我們須在 \LaTeX 指令範圍內變換中文字體。譬如, 中文章節標題通常會使用較大之字體:

```
\section{\ctxfm 耕者有其田}
```

外圈的大括號是 \section 指令的一部分, 內圈的大括號是 $\text{\{ctxfm ...}}$ 中文字體變動的範圍。在此情況下, 下指令時我們僅須使用一圈大括號即可:

```
\section{ctxfm 耕者有其田}
```

6.3.2 舊式中文字體指令

中英文 NFSS 新式字體指令僅適用於 \LaTeX 系統, 原始 \TeX 系統不能使用。如果你的文稿是以 \TeX 指令排版, 仍應使用舊式字體指令。例如, \m12 選用明體 12 點字體, \f11 選用仿宋體 11 點字體, \k17 選用楷體 17 點字體。

在 \LaTeX 文稿中, 你仍可使用舊式字體指令, 但除非萬不得已, 否則請使用新式 NFSS 指令。請注意, 若 \LaTeX 文稿內新舊兩種指令同時使用, 排版時可能出現錯誤。反之, \TeX 文稿內使用新式字體指令, 排版時也會出現錯誤。 \cwtex 會判斷排版文稿是 \TeX 或 \LaTeX , 但有時候可能出錯。執行 \cwtex 時, 你可以特別加上選項要求中文字體指令之轉換格式。若要求使用 NFSS, 選項為 -s :

```
c:\xtemp>cwtex -s test.ctx
```

若不使用 NFSS, 選項為 -f 。

6.3.3 標題與特定段落之中文字體

排版一般中文稿件, 正文通常選用最易閱讀之明體字, 特定段落則選用其他字體以示強調。所謂特定段落, 不外乎文稿題目、章節與表格標題、引用其他文章之段落等。在 \LaTeX 中, 這些都是以指令或指令環境排版。為方便選用中文字體, \cwtex 提供設定中文字體之簡單巨集指令。

如第2章例5 (頁32) 所示, 欲設定文稿 `\section` 與 `\subsection` 標題內使用粗黑中文字體, 我們可在全文設定區下指令:

```
\ctxfdef{\section}{\ctxfbb}
\ctxfdef{\subsection}{\ctxfbb}
```

以上指令將全文每一節與小節之字體以粗黑體排版, 字體大小則由 \TeX 控制。同理, 如果文稿之題目是以 `\chapter` 指令編排, 而題目內之中文字要選用仿宋體, 指令為:

```
\ctxfdef{\chapter}{\ctxff}
```

以上之 `\ctxfdef` 中文字體指令尚可加入選項以控制目錄字體, 請見 7.5.4 節 (頁119) 之說明。

事實上, `\ctxfdef` 可用定義任何指令內之字體。前面 6.2.2 節 (頁82) 曾舉例說明, 如何定義 `\textsl` 英文字體指令內之中文字體。即使是自行定義之巨集指令, 我們也可以使用 `\ctxfdef` 指令以選用中文字體。

除了指令之外, \TeX 另外提供指令環境 (environment) 以排版特別的段落。例如, 排版文稿摘要可使用 `abstract` 指令環境, 排版引文可使用 `quote` 指令環境。若某指令環境內要以特定之中文字體編排, 指令如下:

```
\ctxfdef{quote}{\ctxfk}
...
\begin{quote}
莎士比亞的作品中 ...
...
\end{quote}
```

本例中, 我們以 `quote` 指令環境排版引文, 全文設定區則以 `\ctxfdef` 指令選用楷體字排版其內容。

在 \TeX 中, 我們可以自行定義巨集指令或指令環境, 而巨集指令或指令環境內可包含其他的巨集指令。但是, 要特別注意的是, c\TeX 之字體指令之功能較有限, 我們不能把 `\ctxfdef` 指令包含在 \TeX 的巨集指令或指令環境內。有關於巨集指令之定義方法, 請見 14 章 (頁277) 之說明。

6.3.4 調整中文字距

有些人抱怨 cwTeX 內定之中文字距太小, 排版時常利用完整字體指令加大字距。事實上, 自從文書處理軟體普及之後, 中文排版的最大問題是字距太大, 行距太小。結果使得排版文稿變得難以閱讀。在你動手改變字距之前, 請收集幾本品質較佳的雜誌, 看看其字距與行距之安排, 想想其中的道理。

要改變中文字距, 最簡單的方法是在執行 cwtex 時加入選項。譬如, 中文字距要加大 0.5 點, 中文與數字之間要加大 0.8 點, 中文句點之後空白要加大 1.2 點, 執行指令時之選項為:

```
c:\xtemp>cwtex -z+0.5 -Z+0.8 -zZ+1.2 test
```

請注意, 選項數字之前須加上 + 號。若要縮小字距, 則加入 - 號。

除了調整字距之外, 中文字還可以上下移動, 以配合一些特別的英文字體。要把文稿內全部的中文字 (含中文標點符號) 下移 0.5 點, 執行 cwtex 時應加入選項 -10.5 :

```
c:\xtemp>cwtex -10.5 test
```

如果是要上移中文字, 選項數字應為負值, 如 -1-0.3 。

以上之字體指令雖然可改變中文字距, 但僅適用於調整整篇文稿或某個段落之字距。如果只是想改變章節或表格標題之中文字距, 可使用 $\backslash\text{csp}$ 或 $\backslash\text{cspp}$ 巨集套件, 其定義與應用請見 10.2.4 (頁 195) 節之說明。

此外, 排版中文大字標題時, 有時候須調整某兩個字之間距。譬如, 若以 40 點之仿宋字體排版「排版系統」四個字, 我們會發現「版」與「系」兩字之間距顯得太大。欲調整某兩個字之間距, 可使用 TeX 的 $\backslash\text{kern}$ 指令。譬如, 以下指令:

```
排版\kern-2pt 系統
```

可將「版」與「系」兩個字之間距縮小 2 點; 若取用正值, 間距將加大。

6.4 設定行距

要排出高品質文稿, 除了選用適當字體之外, 字距與行距的選擇也很重要。英文排版的原則是單字 (words) 應盡量靠近; 而行距應大於字距。心理學的視覺研究

發現，人們在閱讀英文文章時，眼睛注視的並不是一個一個的字母，而是整個單字，或幾個單字合併而成的詞。如果單字的距離太大，眼睛移動不順暢，閱讀的速度將受影響。

一般英文排版的原則是：若正文選用 10 點字體，行距則設為 12 點，亦即字級的 120%。若正文使用 12 點的字體，行距則設為 14 點或 14.5 點。不過，這項規則只供作參考，並非一成不變的定則。事實上，有些字體本來就設計得比較大，因此使用不同的字體時，即使字級相同，行距可能也須改變。另外一個重要的考慮因素是行寬。一行的寬度越大，行距通常也須加大。否則，閱讀者的視線從上一行的末端，不容易找到下一行的開頭。

中文字筆劃較英文字複雜，因此中文行距的設定不能完全依照英文的原則。首先，中文字體的高度和寬度和英文/數字並不相同。cwTeX 的英文與數字直接取用 TeX 之字體，仔細比較之後，你會發現中文字高度比英文字母的最高點還高一些，底部則略低於基線。因為中文字較高而且筆劃複雜，如果行距仍設為字體點數的 120%，版面會擠得密密麻麻；因此，中文排版的行距應大於英文稿。至於大到什麼地步才適當，目前似乎也沒有定則，排版者應該用自己的眼睛去判斷。改變行距只需一道指令。因此，我們儘可以去嘗試不同的行距，選取自認為最適當者。本書正文選用 10.95 點字體，行距則設為 17.6 點。

6.4.1 選用什麼字體？

字體選擇是排版首要考慮之一。我們從英文排版文獻整理一些原則，以供參考。這些意見或許不能直接用於中文排版中，不過，舉一反三我們還是可以從中得到一些有用的提示。

首先是有關於字體的概念。一般的報章雜誌，幾乎全部是以明體字編排。英文書籍或報紙也絕大部分是以羅馬字族排版。為什麼呢？西方心理學者的實驗研究發現，羅馬字族的特點是易讀 (legible)。在專業字體設計的術語中，字族可大略區分為 serif (裝飾邊) 及 sans serif (無裝飾邊) 兩種。英文羅馬字族和中文明體字一樣，都是屬於 serif 型態，其特徵是在每一筆劃的尾端有特別的勾勒。因為這項特徵，閱讀時字母較容易分辨，字母容易接續起來形成單字。而且，讀者容易從字母的上半部分辨識出該字母。因為易讀，這種字體普遍使用於報章、書籍與雜誌的排版中。

相對於 serif 字族的,就是所謂的 sans serif,其特徵是筆劃粗細較一致。譬如,英文字體中的 sans serif 及 typewriter 字體;或者中文字的黑體及圓體。在法文中, sans 表示「沒有」,因此 sans serif 的意思是筆劃尾端沒有特別的勾勒。相對於 serif 字族而言, sans serif 字族較不易辨識,因此也較不適用於排版文稿的正文。但是,這種字體很醒目,適合用於強調某段文字,也適合用於排版章節的標題。

排版中英文夾雜的文稿時,請注意字體搭配之問題。譬如,如果中文使用明體或仿宋體,英文應使用 serif 字體。反之,若中文使用圓體或黑體,英文以使用 sans serif 字體為宜。底下第一行文字為明體加上 serif 英文字體;第二行為圓體加上 sans serif:

中文明體字加上英文 serif 字體

中文圓體字加上英文 sans serif 字體

如果將中文明體字於與英文 sans serif 字體共用,排版結果看來並不相稱。

在英文或其它西方國家的文字中,每一套字體中的字母數目大都不超過 256 個,因此專業的字體設計家設計出數以千計的字體可供選用。相對的,常用的中文字就有三、四千字,設計一套幾千字的中文字體要花費相當大的工夫。因此,中文字體的選擇就相當有限。但不管是中文或英文,正文內容和章節標題的字體如何搭配是一門學問。在純英文稿中,雖然可供選用的字體數以千計,但專家的建議是,書籍一頁版面上不應超過三種字體。

排版的目的是把作者的意見清楚、扼要地傳達給讀者。一個版面中使用太多的字體,讀者的注意力分散在花花綠綠的字體上,作者的觀點反而無法有效的傳達。初學排版者往往傾向於使用多一點的字體,這是應該避免的。

TeX 的排版能力強,品質甚佳,但如果使用不當,可能排出慘不忍睹的版面。有興趣於排版者,請參考 [Bringhurst \(1996\)](#) 與 [Taylor \(1994–1995\)](#)。以下謹列出一些常見的錯誤,請儘可能避免之。

- 行距太小

行距太小,版面文字密密麻麻,讀者閱讀時壓力太大。適當的行距應該是多
少?若是排版純英文書籍,直接使用 TeX 之內定值,效果不錯。但若是中文
稿件,因為中文字筆劃複雜,行距須加大。行距與字級有關。字體小者,行

距可以小一些。行距與行長(版面寬度)也有關係。行長較大,行距須加大。版面太寬時,閱讀壓力也上升。行距沒有標準數值。如果是一般文稿,正文選用12點字體,行距請試用18–20點。本書選用11點字體,行距為16.5點。

- 行長太大

行長太大,閱讀時壓力大,其道理與行距太小一樣。如果排版結果要印在A4紙張上,紙面寬度為21公分。很多人把行長設為17公分,左右各只留2公分的空白。事實上,左右兩邊至少應各留3–4公分的空白。如果左右各留4公分,行寬變成13公分,閱讀時會感覺更為順眼。絕大部分的英文雜誌,如The Economist或TIME等,都是以2–3欄型式編排,其道理就是在減少行長。另外,看看國內的報紙、雜誌,我們也會發現每一行的長度都不大。

- 字距太大

早期簡陋的文書處理軟體常把每一個字排得斗大,文字之間的空格也加得很大。研究人員的實驗發現,人在閱讀時,看的並非以一個文字為單位,而是以詞句為單位。字距太大,詞句的長度也增加,眼睛感到吃力,吸收能力也下降。很多人認為`cwTeX`所設定的字距太小,有興趣者不妨自行實驗,同一文稿分別以內定字距與加大字距排版,再比較結果,看看哪一種字距最適合閱讀。

- 正文選用不當的字體

文稿章節標題之字體通常不同於正文之字體。一般而言,章節標題字體的要點是顯目;正文字體之選擇要點是易讀。我們經常可看到一些以楷體、隸書體、仿宋體等排版正文之文稿。事實上,這些字體的特徵是醒目,而非易讀。翻開報紙或任何較具水準的雜誌,內文字體毫無例外都是明體,原因是明體字是最易讀的字體。因此,除非文稿簡短或性質特殊,選用非明體字排版正文之前,請三思而後行。

6.4.2 避頭點

仔細觀察英文書籍,我們發現逗點、句點等標點符號不會出現在一行之首。在專業排版中,這稱為「避頭點」。傳統的中文排版也有類似的作法,其中的道理並不難理解。標點符號是作者議論或語氣停頓之處,也是讀者眼睛稍微休息時。就一本書的版面來看,讀者的眼睛從上一行之尾端轉到下一行開頭時,他預期小停頓

之後有一個新的起頭。因此, 如果一行之首竟然碰到標點符號, 顯然不合讀者的直覺與預期。這是為什麼排版時要「避頭點」的原因。不幸的是, 很多中文桌上排版系統, 都沒有把這項因素納入考慮。

有些排版系統所使用的中文字及標點符號橫寬一致。排版之後, 每一行的中文字上下對得整整齊齊的, 但卻無法避頭點。有些人甚至認為在橫排的版面中, 文字上下對齊才好看。但是, 好的排版是要使文章或書籍容易閱讀。在橫排的書籍或文章中, 讀者並不會從上往下看一本書。因此, 中文字上下對齊並無任何意義。cwTeX 原則上可以處理避頭點的問題, 但偶而仍會有標點符號出現於一行開頭。遇有此種情況, 請將前一兩行的文字增減一字, 以解決避頭點的問題。

7 文稿結構

使用 \LaTeX 排版, 首先須了解文稿結構 (document structure) 的概念。文稿結構是指書籍或文稿各部分的組合方式。一本書的組成部分包含題目頁、版權頁、目錄、序言、章節、附錄、索引等。相對而言, 一般的短文稿通常只有題目、作者、日期、章節、參考文獻等。就題目而言, 書籍的排版方式與一般文稿之排版方式可能不同。書籍通常把題目排為單獨的一頁, 一般文稿則可能把題目直接排版於第一頁的上方。

排版時, 除了設計文稿各部分之版面外, 我們須注意: 同一版面性質之文字應採相同之排版。譬如, 各章之標題應採相同之字體及大小; 又如, 各圖表之標題也應採相同格式。在排版長篇文稿時, 這一點特別容易出錯。翻開一本長達數百頁的書籍, 我們有時會發現, 前面部分節標題是以 14 點圓體字排版, 到了後面卻變成 15 點的粗黑體。

\LaTeX 排版系統採用文稿結構的概念, 並利用巨集套件 (package) 與巨集指令 (macros) 處理上述的問題。前兩章已說明排版指令的意義, 所謂「巨集指令」是指把數個或數十個指令結合在一起, 以進行某項排版工作。譬如, \LaTeX 排版節之指令為 `\section`。下達此一指令之後, 文稿須結束上一段文字, 空出一點距離, 以特定之字體排出標題文字, 之後再空出一點距離, 準備排版底下文字。以上每一個動作可能都需數個 \TeX 基本指令, 為方便起見, \LaTeX 把所有的指令集合在一起, 並命名為 `\section`, 這就是一個巨集指令。進一步而言, 我們可以把性質相關的巨集指令組合在一起, 這就是巨集套件。使用巨集指令的好處是不需輸入眾多的複雜基本指令, 減少發生錯誤。

開始動手排版之前, 我們首先決定採那一種文稿結構。最常用的文稿結構有兩種, 第一種是排版較短的文稿用的, 另一種是排版長篇文稿或書籍用的。一旦選定文稿結構, \LaTeX 即以內定之格式排版。不過, \LaTeX 所提供的文稿結構你可能不滿意。如果不喜歡內定之版面設計, 我們也可以自行設計。即使 \LaTeX 所提供之指令不能滿足你的需求, 我們仍然可以透過其他人所寫的巨集套件達到重

新設計的目的。

每本書的內容不同，故有其特別的版面設計需求。以本書為例，我們須舉一些小例子說明指令用法。為保證前後各個例子都以相同的格式出現，我們儘可能使用巨集指令來排版這些例子。使用巨集指令的最大好處如下：在排版中途，我們可能覺得原先設計的版面並不理想，希望更改設計。譬如，一開始我們設計以 11 點字體排版例子，後來改變主意，希望改用 10 點之字體。如果是用巨集指令排版這些例子，只要調整巨集指令之定義內容，即可一舉更正每一個例子之字體。如果不是透過巨集指令，就必須一個一個例子修改，不僅麻煩，也容易出錯。

如果僅是排版一般的論文，則 \LaTeX 所提供之指令已足敷所需。但如果你對於排版品質要求較高，則花一點時間了解某些巨集套件之使用，將會大幅提升排版品質。在 CTAN 網站上有許多現成的巨集套件可供下載使用。若能善用網路上的資源，再怎麼特別的排版需求幾乎都可以找到現成的巨集套件來解決。本書以下各章除了說明 \LaTeX 的指令之外，也將介紹一些常用的巨集套件。

7.1 長度與寬度

書籍或文稿版面設計的第一步是決定版面大小。簡短的文稿通常列印於 A4 或 letter size 紙張上。在 \LaTeX 中，一旦選定紙張尺寸，版面大小即有內定值。當然，我們也可以自行設定版面的尺寸。由第 2 章的例子可知，任何 \LaTeX 文稿至少含有下列三道指令：

```
\documentclass{article}
\begin{document}
...
\end{document}
```

第一行指令 `\documentclass` 稱為「文件類別指令」，這通常是文稿的第一道指令。本例中，文件類別指令之後接著 `article` 選項，設定以短文格式排版。若是排版書籍，我們可選用 `book` 選項。

本例中另外兩行指令 `\begin{document}` 與 `\end{document}` 是成對出現的，兩道指令之間即為排版文稿內容。後一道指令是文稿最後一行指令；其後的任何文字或指令 \LaTeX 皆不處理。此種成對出現的指令所涵蓋的範圍稱為「指令環境」(environment)。本例之指令即稱為 `document` 指令環境。 \LaTeX 提供許多指令

表 7.1: 常用的長度單位

cm : 公分	pc : pica = 12 點
mm : 公厘 = 0.1 公分	bp : big point = 1/72 吋
in : 吋 = 2.54 公分	em : 約為大寫字母 M 之寬度
pt : 點 = 1/72.27 吋	ex : 約為小寫字母 x 之高度

環境, 譬如排版表格可使用 `tabular` 指令環境。

7.1.1 長度單位

版面設計的第一步是設定版面的寬度與高度、訂定行距等等。所有這些工作都須使用長度的單位。此外, 字體大小也是長度的一種。專業字體設計家在設計字體時, 都設想實際排版時最美觀的大小, 稱之為設計尺寸 (design size)。L^AT_EX 的英文及數學符號字體, 大部分的設計尺寸都是 10 點 (printing points, 簡稱為 pt)。1 點等於 1/72.27 吋, 或大約 0.0351 公分。簡單來說, 英文字體 10 點之大小大約是各字母中的最高點 (如字母 h 之頂端) 到最低點 (如字母 y 之底) 的距離。為了與英文字的大小配合, c_wL^AT_EX 中文字體的設計尺寸也都是 10 點。專業排版常以 printing point 作為長度的單位, 但我們也可以使用一般的長度單位。表 7.1 列出常用的長度單位。

設定長度的方法是在指令之後直接以等號定義, 例如:

```
\textwidth=20cm
\textwidth20cm
```

如果更精簡, 等號可以省略, 如第 2 行指令。事實上, L^AT_EX 設定長度的標準指令是 `\setlength`。要把版面寬度定為 20 公分, 指令如下:

```
\setlength{\textwidth}{20cm}
```

以上所設定的長度稱為固定長度 (fixed length), 有時候我們必須設定所謂的彈性長度 (rubber length)。舉例來說, 排版時我們須設定版面的高度與行距。版面高度通常是固定長度, 如果行距也是固定, 可能出現的情形是版面最底下無法剛好把最後一行擺進去。如果行距設為彈性長度, 萬一最後一行擺不進去時, L^AT_EX 可以把行距都縮小一些, 使最後一行可以排入。底下是一個例子:

```
\setlength{\textwidth}{20cm}
\setlength{\baselineskip}{15pt plus0.4pt minus0.2pt}
```

第 2 行指令設定行距為一彈性長度, `\baselineskip` 為行距之指令, `15pt` 為行距之正常大小, `plus0.4pt minus0.2pt` 為彈性大小。因此, 行距最大可以是 `15.4pt`, 最小是 `14.8pt`。

`\setlength` 指令用以直接設定長度, 但某些變數已有內定值, 如果只是要加長或縮短, 可以使用 `\addtolength` 指令。譬如,


```
\addtolength{\baselineskip}{10pt}
```


可將標準行距加大 `10pt`。若 `10pt` 改為 `-5pt`, 標準行距將減小 `5pt`。

以上介紹的是長度單位及設定方法。有時候我們須在文稿內產生水平或垂直空白。產生垂直空白的指令為 `\vspace`, 相對而言, 橫向空白之指令為 `\hspace`。譬如, 要在版面中空出垂直 3 公分, 指令為 `\vspace{3cm}`; 同理, 要在某行文字內加入水平 2 公分的空白, 指令為 `\hspace{2cm}`。請注意, `\vspace` 指令的位置若恰好是在一頁版面的最上端, 指令將失效。但如果是使用 `\vspace*` 指令, 不管是下於何處, 指令都有效。`\hspace` 也有類似的情況; 指令若恰位於一行之最左端, 指令也是無效。若使用 `\hspace*` 指令形式, 則下於任何地方都有效。

以上控制間距之指令須在大括號內加上間距之長度, 這是所謂的強制變數 (mandatory argument)。相對而言, 有一些指令除了強制變數之外, 還可加入選項變數 (optional argument)。強制變數是加於大括號內, 選項變數則加於中括號內。顧名思義, 選項變數可加可不加。譬如, \TeX 劃直線的指令為:

```
\rule[ lift]{width}{height}
```

以上指令中, `width` 設定直線長度, `height` 設定線的粗細, 這兩個都是強制變數。相對而言, `lift` 屬於選項變數, 設定直線往上抬高的距離。因此, 要畫出長度 1 公分, 粗細 0.1 公分的直線: , 指令為 `\rule{1cm}{0.1cm}`。

如果要把直線抬高 0.1 公分, 可以加上選項: `\rule[1mm]{1cm}{0.1cm}`, 畫出之直線為: .

7.1.2 設定版面大小

動手排版之前應先設計好整篇文稿的版面。文件類別指令的目的即是在選定最

表 7.2: 紙張尺寸

a4paper	21 × 29.7 公分	letterpaper	8.5 × 11 吋
a5paper	14.8 × 21 公分	legalpaper	8.5 × 14 吋
b5paper	17.6 × 25 公分	executivepaper	7.25 × 10.5 吋

基本的版面設計。在文件類別指令中加入選項, 即可選用特定的版面尺寸或字級。對 \LaTeX 而言, 文稿版面之組成元素如圖 7.1 所示。台灣目前同時通行美國與歐洲的紙張規格。美國常用的紙張尺寸是 letter size 與 legal size; 歐洲或日本通用的尺寸則是 A4, A5, B5 等。如果不作選擇, \LaTeX 自動選用 letter size, 其大小為 8.5 × 11 英吋。

選擇紙張規格很簡單, 只須在 `\documentclass` 指令中加入選項即可。紙張選項如表 7.2 所示。若選用 A4 紙張排版短文 (`article`), 字體選用 12pt, 指令為:

```
\documentclass[12pt,a4paper]{article}
```

`\documentclass` 指令之後的方括號內為選項, 若不加選項, \LaTeX 將使用內設值。因此, 若指令為:

```
\documentclass{article}
```

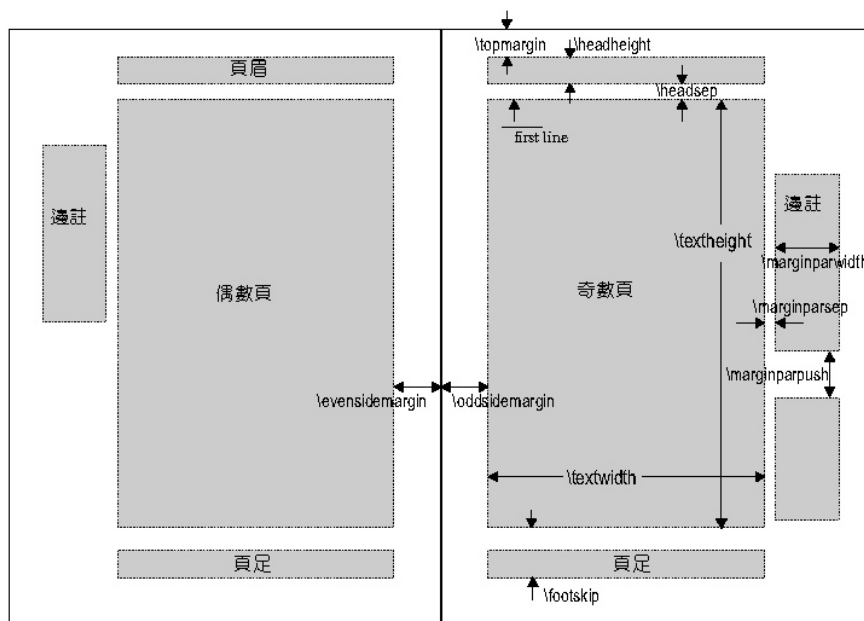
\LaTeX 將選用 letter size 紙張, 並以 10pt 字體排版。大括號中之選項並無預設值, 因此一定要加上。

如果要使用特別的紙張尺寸, 須自行設定。文稿的版面設計必須在正文之前決定, 因此紙張尺寸之指令須置於 `\begin{document}` 之前, 此一區域稱為「全文設定區」(`preamble`)。譬如, 若選用的版面 (含文字與四周之空白) 高 26 公分, 寬 20 公分, 須在全文設定區加入下列兩行指令:

```
\paperheight=26cm
\paperwidth=20cm
```

紙張大小選定之後, 版面正文方格及其他部分之規格都會隨之調整。

如圖 7.1 所示, 文稿的每一頁除了正文文字之外, 還有頁足、頁眉、邊註等。在 \LaTeX 中, 排版於正文文字方格上方之資訊稱為 header, 傳統中文排版稱之為「天頭」, 本書將稱之為「頁眉」; 相對而言, 正文文字方格下方之資訊稱為 footer, 傳統



<code>\textwidth</code>	正文方格寬度。	<code>\marginparsep</code>	正文方格與邊註的距離。
<code>\textheight</code>	正文方格高度。	<code>\marginparwidth</code>	邊註的寬度。
<code>\linewidth</code>	正文一行之寬度。	<code>\marginparpush</code>	連續兩個邊註之間的最小垂直距離。
<code>\headheight</code>	頁眉高度。	<code>\columnseprule</code>	正文有兩欄或以上時，兩欄間分隔直線的寬度。
<code>\topmargin</code>	頁眉上方之空白。	<code>\columnwidth</code>	正文有兩欄或以上時，每一欄的寬度。
<code>\columnsep</code>	正文有兩欄或以上時，兩欄間的距離。		
<code>\oddsidemargin</code>	取 <code>twoside</code> 選項時，奇數頁左邊的空白。若未取 <code>twoside</code> 選項，此項設定值即為各頁左邊的空白。		
<code>\evensidemargin</code>	取 <code>twoside</code> 選項時，偶數頁右邊的空白。		

圖 7.1: L^AT_EX 之版面與控制指令

排版稱之為「地脚」，本書則稱之為「頁足」。頁眉與頁足之排版設計，西文排版中稱之為 page style，本書稱之「頁面格式」。正文方格的大小、正文與頁眉的距離、邊註的寬度等，都有內設值，但也都可以重新設定。正文方格是指 `\textheight` 與 `\textwidth` 所形成之長方形，如果要改變其大小，須在全文設定區加入下列兩行指令：

```
\textheight=20cm
\textwidth=16cm
```

以上指令重新設定正文方格之寬度為 16 公分，高度為 20 公分。其他各項距離的設定也以類似方式為之。

前面的 `\paperheight` 指令雖然選擇紙張大小，排版結果可能還是列印於 A4 紙張上。列印時，列印程式將選取白紙左上方的某特定點為開始列印的參考點。參考點通常距離紙張上沿及左沿各一公分。因為正文方格的寬度及高度是自行設定，列印之後正文方格四周的空白可能並不對稱。此時，我們可以利用 `\voffset` 與 `\hoffset` 指令調整之。例如，要將正文方格上移 0.5 公分，右移 1 公分，可以在全文設定區加入下列指令：

```
\voffset=-0.5cm
\hoffset=1cm
```

7.2 文稿結構與文件類別

排版之前，須決定文稿的排版結構。因此， \TeX 排版文稿開頭的第一道指令是選定文稿結構：

```
\documentclass[12pt]{article}
```

其中，`\documentclass` 指令的功能之一是選用排版文件之類別 (document class)，本例之 `article` 則稱為短文文件類別。除了 `article` 文件類別之外，複雜的長篇文稿或書籍可使用 `book` 文件類別排版。此外，我們還可使用 `report` 報告文件類別，其章節結構比 `article` 複雜，但比 `book` 簡單一些。排版投影片可使用 `slides`，排版信函可使用 `letter` 文件類別。

以上之文件類別可以靈活應用。若書籍之內容單純，我們可以直接使用較簡

單的 `article` 文件類別排版。反之，若文章的內容複雜，則以 `book` 文件類別編排可能較方便。如果文稿中含有許多數學式，我們還可以使用 `amsart` 與 `amsbook` 文件類別，這是美國數學學會 (American Mathematical Society) 爲了排版數學文稿所發展出來的，請見第 9 章之說明。

文件類別一經選定，文稿結構同時也決定。舉例言之，若選用 `book` 文件類別排版，則以 `\title` 指令排版書名時，題目文字將自成一頁。反之，若是選用 `article` 文件類別排版，`\title` 指令會直接將題目文字排版於第 1 頁上方，而不是獨立排爲一頁。

7.2.1 指令選項

許多 \TeX 之指令都可以設定選項 (options)。舉例來說，上面所介紹的文件類別指令，除了選用 `article` 文件類別之外，還加上 `[12pt]` 之字體選項。指令選項須以方括號括起來，以有別於以大括號括起來之強制選項。若不加字體選項， \TeX 將以 10 點字體排版正文。

事實上，`[12pt]` 選項除了改變字體大小外，也改變版面其它部分。譬如，註解文字之字體會隨著加大一些；文稿中的數式若有上下標，其大小也會配合正文字體而改變。若嫌 `12pt` 之字體太大，我們也可以選用 `11pt`。

除了選擇字體大小之外，文件類別指令還可以加入許多其他選項。我們選擇其中較重要的簡單介紹如下。首先，紙張的選擇可使用下列選項：

<code>letterpaper</code>	11×8.5 英吋
<code>a4paper</code>	29.7×21 公分
<code>a5paper</code>	21×14.8 公分

其中，`letterpaper` 爲內定值。其次，大多數的排版結果是列印於垂直紙面上，這稱爲 `portrait` (畫像) 模式，也是 \TeX 內定之模式。但若在文件類別指令中加入 `landscape` 選項，則文稿將以橫向或所謂的 `landscape` (風景) 模式排版。

在 `article` 文件類別中，文稿每一頁將排版於紙張相同的位置。但如果選用 `book` 格式，單雙頁將對稱於書脊排版。換言之，紙面上右頁的版面會較靠近左邊；左頁的版面則較靠右邊。本書是以 `book` 文件類別排版，你現在閱讀的這一頁與對面一頁就是對稱於書脊。依原始設定，`article` 與 `report` 文稿的單雙頁版面

都是位於紙面同樣位置。如果希望單雙頁對稱於書脊編排，可在文件類別指令中加入 `twoside` 選項。

以 `book` 文件類別排版書籍時，每一章開頭之標題頁會自動從右頁（單數頁）開始編排。如果要取消此項設定，文件類別指令中應加入下列選項：

```
openany
```

最後，數學式之排版格式也可以用選項控制。依原始設定，數學式將居中編排，數式編號則排於右邊。若數式編號要置於左邊，應在文件類別指令中加入下列選項：

```
leqno
```

如果是數式要靠左編排，選項指令為：

```
fleqn
```

我們還可以進一步使用 `\mathindent` 指令讓靠左編排之數學式從特定位置開始排版。譬如，下列兩行指令：

```
\documentclass[12pt,fleqn]{article}
\mathindent=1.5cm
```

即設定讓每一行數學式從距離文字版面邊緣 1.5 公分處開始排版。

除了以上所述之外，文件類別指令其他選項如下：

```
twocolumn  以兩欄形式編排，
titlepage  使文章標題、作者名字等自成一頁。
```

`twocolumn` 選項雖然可以設定排版兩欄式版面，但使用上限制較多。欲排版多欄位版面，請見 8.10 節之說明。

7.2.2 巨集套件

以上 5 種文件類別所提供之指令雖然已能滿足多數人之需求，但任何事情都有更上一層樓的空間，排版也不例外。因此，各國的 \LaTeX 專家與使用者又寫出許多應付特別需求之巨集套件 (package)。譬如， \LaTeX 本來就提供排版表格之指令，

但是有人又寫了一套功能更強之 `array` 巨集套件。又如, \TeX 中已有指令可以排版頁眉, 但 `fancyhdr` 巨集套件功能更強, 使用更方便。我們如何可使用這些巨集套件呢? 以 `array` 為例, 欲使用巨集套件內之指令, 首先必須在全文設定區以 `\usepackage` 指令引用之,

```
\documentclass[11pt]{article}
\usepackage{array}
```

如果要同時使用兩種以上之巨集套件, 引用指令為:

```
\documentclass[11pt]{article}
\usepackage{fancyhdr}
\usepackage{array}
```

或者

```
\documentclass[11pt]{article}
\usepackage{array, fancyhdr}
```

請注意, 各選項之間不得留有空白。

以上所介紹的文件類別指令名為 `\documentclass`, 這是新版 $\text{\TeX 2}\epsilon$ 之名稱; 舊版 (2.09 版) 稱之為 `\documentstyle`。新版 \TeX 中, 巨集套件是由 `\usepackage` 指令引入; 但在舊版中則直接將之作為文件類別指令之選項。如果你仍使用舊版 \TeX 2.09 , 引用巨集套件之指令為:

```
\documentstyle[11pt, array, fancyhdr]{article}
```

新版的 \TeX 是在 1994 年開始流通。如果你已由舊版轉為新版, 但以往許多的文稿是以舊版指令排版, 這些舊指令絕大部分在新版中仍然可以使用, 不過執行速度會慢一些。

7.2.3 article 文件類別

圖 7.2 是短文的例子, 上半部分為排版結果, 其下為輸入之指令與文字。文稿第 1 行為文件類別指令 `\documentclass`, 本例中, 此道指令設定以短文 `article` 編排文稿, 並以選項 `[12pt]` 選用 12 點字體。若不選字體大小, 內定值為 10 點。第 4 行 `\begin{document}` 指令開始即進入 `document` 指令環境, 其內為文稿內容,

文稿最後一行為 `\end{document}`。本例子之檔案名為 `exmp-art.ctx`，置於 `c:\texmf\cwtex\examples` 檔案夾內，請試自行排版。

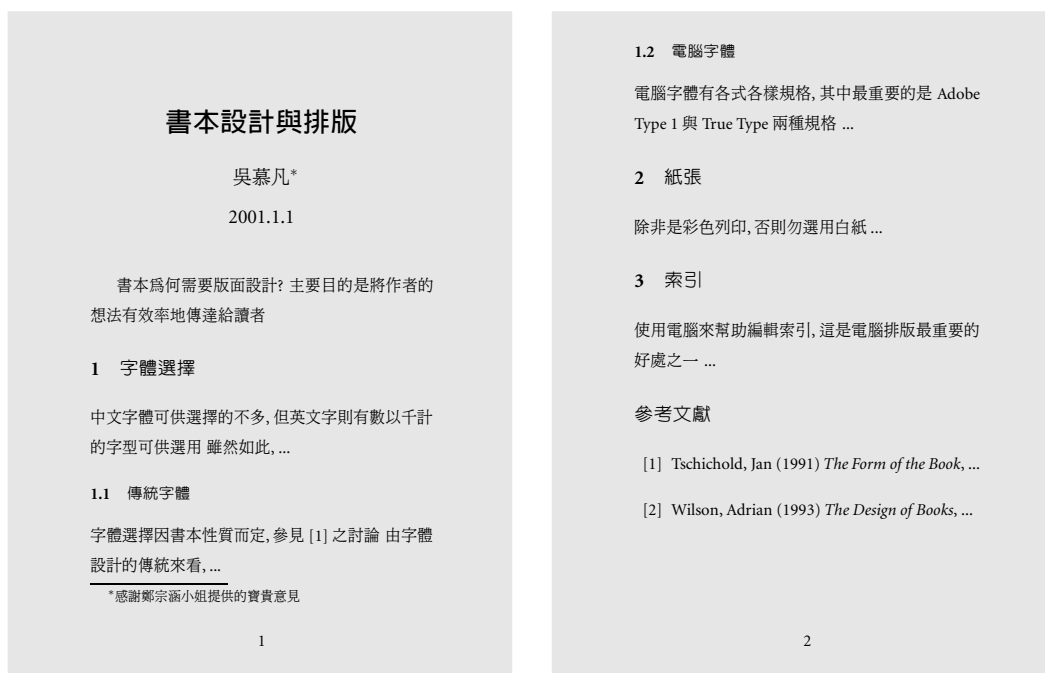
文稿一開始通常先排版文章題目、作者姓名、日期等，這是由 6–9 行指令所排版。第 6 行以 `\title` 指令排版題目，並以 `\ctxfbb` 中文字體指令選用粗黑體。第 7 行以 `\author` 指令排版作者名字。第 8 行之 `\date` 指令若省略， \TeX 將自動填入排版當天的日期。第 11 行 `\maketitle` 指令之作用是指示在此行之前的文字應排版為題目。在 `article` 文件類別下，題目之下即接著排版正文內容；但在 `book` 文件類別下，題目將獨占一頁。

文稿正文之節、小節是以 `\section` 與 `\subsection` 指令編排， \TeX 會自動編上號碼。圖 7.2 中，文稿計分 3 節，都是以 `\section` 指令編排。章節指令除了自動編號之外，還會選用較大之字體。本例中，我們以 `\ctxfr` 選用圓體字排版節標題。選用中文字體之指令為 3–4 行之 `\ctxfdef`。第 21 行與 26 行之 `\subsection` 指令編排小節標題，因為位於第 1 節之內，因此自動編號為 1.1 與 1.2。小節之中文標題也是選用圓體字。

專業文章經常引用參考文獻。如果參考文獻多，徵引時容易出現錯誤。 \TeX 提供 `thebibliography` 指令環境以排版參考文獻，41–46 行指令說明使用方法。輸入徵引文章或書名時皆以 `\bibitem` 起頭，並任取一簡名。例如，第 42 行列出 Tschichold 的著作時，指令為 `\bibitem{tsc}`，其中 `tsc` 為自取之簡名。正文第 23 行引用此著作時，使用 `\cite{tsc}` 指令。排版時， \TeX 自動從參考文獻中找到 `tsc` 項，並在正文中代入其排序號碼。因此項著作為徵引文獻的第 1 篇，排版之後變成“參見 [1] 之討論”。文稿 41 行輸入 `thebibliography` 指令環境時，末端加上 “{99}”，目的是為參考文獻的序號預留 2 位數字之排版空間。如果文稿引用之參考文獻超過 100 篇，則 {99} 應改為 {999}。

在 `article` 文件類別中，使用 `thebibliography` 指令環境時， \TeX 會在參考文獻項目之前加上英文字 **References**。若是中文文稿，我們可以更改此項標題為中文字。在 \TeX 中，**References** 一字是以 `\refname` 指令代表。第 2 行指令在全文設定區使用 `\renewcommand` 指令，將 `\refname` 之內容重新定義為「參考文獻」四字，並選用圓體字。

以上說明如何使用 `thebibliography` 指令環境排版參考文獻。實際上， \TeX 雖然提供排版參考文獻的指令環境，但並不強制非得使用不可；我們也可以用一般的指令排版參考文獻。不過，文件類別指令 `\documentclass` 與 `document` 指令



```

1 \documentclass[12pt]{article}
2 \renewcommand{\refname}{\ctxfr 參考文獻}
3 \ctxfdef{\section}{\ctxfr}
4 \ctxfdef{\subsection}{\ctxfr}
5
6 \title{\ctxfbb 書本設計與排版}
7 \author{ 吳慕凡\thanks{
8 感謝鄭宗涵小姐提供的寶貴意見。}}
9 \date{2001.1.1}
10 \begin{document}
11 \maketitle
12 \fontsize{12}{20pt}\selectfont
13 書本為何需要版面設計?
14 主要目的是將作者的想法有效率地傳達給讀者。
15
16 \section{ 字體選擇}
17 中文字體可供選擇的不多,
18 但英文字則有數以千計的字型可供選用。
19 雖然如此, ...
20
21 \subsection{ 傳統字體}
22 字體選擇因書本性質而定,
23 參見 \cite{tsc} 之討論。
24 由字體設計的傳統來看,
25 ...
26 \subsection{ 電腦字體}
27 電腦字體有各式各樣規格,
28 其中最重要的是 Adobe Type 1
29 與 True Type 兩種規格。
30 ...
31
32 \section{ 紙張}
33 除非是彩色列印, 否則勿選用白紙
34 ...
35
36 \section{ 索引}
37 使用電腦來幫助編輯索引,
38 這是電腦排版最重要的好處之一。
39 ...
40
41 \begin{thebibliography}{99}
42 \bibitem{tsc} Tschichold, Jan (1991)
43 {\it The Form of the Book}, ...
44 \bibitem{wilson} Wilson, Adrian (1993)
45 {\it The Design of Books}, ...
46 \end{thebibliography}
47 \end{document}

```

圖 7.2: 短文 article 之文稿結構

環境等三道指令是一定要有的。

7.2.4 研究生論文

除了一般的研究論文之外，國內的研究生須寫一畢業論文。如果你有機會在圖書館查閱這些論文，不難發現每一篇論文的排版各有特色，充分表現排版者的想像力。有些人用 14 點字體編排正文，甚至更大的字體。不少人喜歡以楷體作為正文字體，有人甚至使用隸書體或古印體排版正文。結果慘不忍睹！最常見的問題是：行寬太大，行距則太短！

排版任何文稿，首重精確，其次是易讀；學術論文也不例外。以 \LaTeX 排版研究生論文，我們可使用 `article` 文件類別，或者 `book` 文件類別。本小節將介紹一個以 `article` 文件類別排版碩士論文的例子，下一小節則說明同一論文以 `book` 文件類別編排。在 `cwTeX` 例子檔案夾內有 `thesis1.ctx`，這是由一篇實際的碩士論文稍加編輯而成。全篇論文的結構請見圖 7.3，其中所使用的指令以下各章將詳加說明，此處僅簡單介紹指令的大要。

- 第 1 行：選定以 `article` 文件類別排版。
- 2–4 行：引入所需之巨集指令，其中第 2 行的 `titlesec` 巨集套件設定節與小節標題使用 `sans serif` 字體，並且略為縮小標題與上下文之間距。因為節與小節標題之英文與數字選用 `sans serif` 字體，故標題之中文字選用粗黑體，這是 9–10 行指令作用。
- 6–7 行：設定文字版面之大小與位置。
- 12 行：將圖標題由 “Figure 2” 改為 “圖 2”，第 13 行則將表標題由 “Table 5” 改為 “表 5”。
- 16–17 行：定義一指令 `\laref`，用以排版參考文獻，排版方法如第 95–98 行所示。在 `thesis1.ctx` 檔案中，我們還定義了其他指令，其中之一用於使表格中之數字上下對齊，另一個則是設定將註解的行距稍微加大。
- 20 行：`\begin{document}` 指令表示論文開始。
- 23–40 行：排版題目頁，包括校名、論文題目、作者名字、日期等。
- 42–47 行：排版特別獻詞 (`dedication`)，或向特別單位或人致謝之文字；

```

1 \documentclass[12pt]{article}
2 \usepackage[sf,small]{titlesec}
3 \usepackage{array,dcolumn,psfrag,graphicx}
4 \usepackage{booktabs,lscapex}
5
6 \hoffset=1cm\voffset=-1cm
7 \textwidth=12.5cm \textheight=22cm
8
9 \ctfdef{\section}{\ctfbb}
10 \ctfdef{\subsection}{\ctfbb}
11
12 \renewcommand{\figurename}{ 圖}
13 \renewcommand{\tablename}{ 表}
14 \renewcommand{\contentsname}{\ctfbb 目錄}
15
16 \newcommand\leref{\smallskip\noindent%
17 \hangindent=\parindent}
18 ...
19
20 \begin{document}
21 \fontsize{12}{21pt}\selectfont
22 % title page
23 \thispagestyle{empty}\vspace*{1cm}
24 \fontsize{14.4}{25pt}\selectfont
25 \begin{center}
26 國立台灣大學經濟學研究所\
27 碩士論文\
28 ...
29 \end{center}
30
31 \vspace*{2cm}
32 \begin{center} \Large
33 嘉南大圳之經濟效益分析
34 \end{center}
35
36 \vfill
37 \begin{center}
38 研究生: ...
39 1997 年6月
40 \end{center}
41
42 \newpage
43 \thispagestyle{empty}
44 \vspace*{.1\textheight}
45 \begin{center}\large \ctfbb
46 感謝嘉南農田水利會 ...
47 \end{center}
48
49 \newpage
50 \thispagestyle{empty}
51 \begin{center}\Large \ctfbb
52 謝詞
53 \end{center}
54 \fontsize{12}{21pt}\selectfont
55 \bigskip
56 \noindent
57 三生有幸, 能在台大經研所 ...
58
59 \newpage
60 \thispagestyle{empty}
61 \begin{center}\Large \ctfbb
62 論文提要
63 \end{center}
64
65 \bigskip\noindent
66 日治時期的文獻多認為,
67 ...
68
69 \newpage
70 \tableofcontents
71
72 \newpage
73 \section[ 前言]{前言}
74 在日治時期, 米穀產業與 ...
75
76 \newpage
77 \section[ 嘉南大圳]{嘉南大圳}
78 嘉南大圳以遠闊之灌溉面積 ...
79
80 \subsection[ 灌溉區域]{灌溉區域}
81 嘉南大圳橫跨台南州 ...
82 ...
83
84 \newpage
85 \section[ 結語]{結語}
86 本文透過推論與計算, ...
87
88 \newpage
89 \addcontentsline{toc}{section}{ 附表}
90 ...
91
92 \clearpage
93 \addcontentsline{toc}{section}{ 參考文獻}
94 \section*{ 參考文獻}
95 \leref Greene, William H. (1993), ...
96
97 \leref 三浦博亮 (1923),
98 (嘉南大圳 輪作方式), ...
99
100 \end{document}

```

圖 7.3: 以 article 文件類別排版論文

- 49–57行: 排版一般的致謝詞。以上各頁的排版指令, 第一道都是 `\newpage` , 其次為 `\thispagestyle{empty}` , 後一指令的目的是避免頁碼出現。一般英文書的標題頁都不排版頁碼, 本例子即照此規範。
- 59–67行: 排版論文摘要, 標題「論文提要」四個字是以粗黑體排版。
- 70行: `\tableofcontents` 指令用以排版目錄。E_TX 自動將節 `\section` 與小節 `\subsection` 的標題排入目錄, 目錄上端則加上“Contents”英文字。前面第 14 行指令的目的則是將“Contents”改為中文「目錄」兩字, 並選用粗黑體。論文中某些特別段落的標題若非以 `\section` 或者 `\subsection` 指令編排, 則標題不會自動納入目錄內。例如, 本例子參考文獻之標題是以 `\section*` 指令編排, 因為加上 * 號, E_TX 不自動編號, 也不納入目錄內。故我們須以 `\addcontentsline` (第 93 行) 指令將標題編入目錄。第 89 行指令也是同一作用。
- 72–86 行是論文內容之排版。本論文例子計有 4 節, 每一節都以 `\section` 指令開頭。第 1 節指令為:

```
\section[ 前言]{前言}
```

其中, 大括號內之標題文字將排版於節標題處; 中括號內為指令選項, 可以省略不加。若不加入選項, 則目錄及頁眉之排版即取用大括號內之標題文字。反之, 若加入中括號選項, `cwTex` 即取用其中之字體排版。前面第 9 行已使用 `\ctxfdef` 指令設定節標題內之中文字以粗黑體編排, 故 {前言} 內不須再加上中文字體指令。排版目錄之 [前言] 也未加入中文字體指令, 故 `cwTex` 自動選用內定之明體字。如果目錄要以其他字體排版, 則第 9–10 行指令須修改。例如, 若目錄內之中文字改用仿宋體, 指令如下:

```
\ctxfdef{\section}[\ctxff]{\ctxfbb}
```

- 93–99 行排版參考文獻, 採用 `hanging indent` 格式排版。E_TX 並無現成的指令可用, 故我們自行定義 `\laref` 指令, 請見 16–17 行。

依 E_TX 設計, 在 `article` 文件類別下, 節標題並不新起一頁。本篇論文之 4 個主要部分是以 `\section` 指令排版, 我們在指令之前加上 `\newpage` 新起一頁,

以強化區分效果。

7.2.5 book 文件類別

上一小節的例子以 `article` 文件類別排版論文，同樣的論文也可以使用 `book` 文件類別排版，但部分指令須稍加修改。以 `article` 文件類別排版時，整篇論文分為 4 節，若改以 `book` 文件類別排版，則分為 4 章；章標題之格式與原來不同。此外，每一頁之頁眉 (header) 或頁足 (footer) 會自動列出章節標題。我們將以 `book` 文件類別排版之例子檔案命名為 `thesis2.ctx`，置於 `cwTeX` 例子檔案夾內以供參考。為方便說明，主要的修正指令置於圖 7.4。

- 第 1 行：選用 `book` 文件類別。我們另外加上 `oneside` 及 `openany` 兩選項。依 `TeX` 內定值，以 `book` 文件類別排版時，左頁 (偶數頁) 將往右靠向書脊，右頁 (奇數頁) 則將往左靠向書脊。但一般的研究生論文都是單面裝訂，以上的版面移動反而使裝訂產生困擾，故選用 `oneside` 設定，取消靠向書脊之設定。另外，以 `book` 文件類別排版時，每一章的標題頁都由單數頁開始編排，因此，標題頁之前一頁可能是完全空白的。在單面裝訂時，此一設定也無意義，故我們加入 `openany` 選項，設定章標題可以在奇數頁或偶數頁開始。
- 第 3 行：選用 `fancyhdr` 巨集套件以設定頁眉與頁足。實際的設定指令是在 8–14 行，指令內容之說明請參考 17.4 節 (頁 338)。本例中，我們設定頁眉右端排版章標題。但請注意，論文末端的附表及文獻索引並非以 `\chapter` 指令排版，故我們須重新以 `fancyhdr` 之指令控制，請參見 `thesis2.ctx`。
- 第 5 行以 `\hoffset=1cm` 指令設定版面向右移動 1 公分；`\voffset=1cm` 指令則設定向下移動 1 公分。第 6 行指令設定版面之尺寸。
- 16–17 行：以 `titlesec` 巨集套件之指令重新設計章標題。章標題分兩行，第 1 行為「第 x 章」，第 2 行為標題文字。指令內容之說明請參考 17.2 節 (頁 326)。
- 19–20 行：`cwTeX` 字體指令，選擇以粗黑體排版章標題。

就本例子而言，使用 `article` 文件類別與 `book` 文件類別的主要差別是在章標題的排版上。以第 1 章為例，前者排版為「1. 前言」，後者排為兩行，第 1 行為

```

1 \documentclass[12pt,oneside,openany]{book}
2 ...
3 \usepackage{fancyhdr}
4
5 \hoffset=1cm\voffset=-1cm
6 \textwidth=12.5cm \textheight=22cm
7
8 \pagestyle{fancy}
9 \fancyhf{}
10 \renewcommand{\chaptermark}[1]{\markboth{%,
11 第~\thechapter~ 章 ~ #1}-}}
12 \fancyhead[RO,RE]{\leftmark}
13 \fancyfoot[CO,CE]{\thepage}
14 \renewcommand{\headrulewidth}{0pt}
15
16 \titleformat{\chapter}[display]{\centering\LARGE\sf}
17 {\ctxfbb 第\thechapter\ 章}{0.2cm}{}
18
19 \ctxfdef{\chapter}{\ctxfbb}
20 \ctxfdef{\section}{\ctxfbb}

```

圖 7.4: 以 book 文件類別排版論文

「第 1 章」, 第 2 行為「前言」, 兩行都居中編排。實際上, 利用 `titlesec` 巨集套件之設計指令, 我們也可以在 `article` 文件類別下排版出後者之標題格式。因此, 若能充分利用巨集套件的話, 不同的文件類別可以排出相同的版面。當然, 若文稿甚長時, 使用 `book` 文件類別仍有其便利之處。第 18 章將有進一步的說明。

7.3 文稿題目

一般書籍或文章的題目頁包含題目、作者、出版日期、與謝詞 4 個部分。以 `book` 或 `report` 文件類別排版時, 題目將獨占一頁; 但在短文 `article` 中, 文稿題目區域之後即緊接著排版正文 (或摘要), 題目並未獨占一頁。不過, 為了簡化文字說明, 以下都以題目頁稱之。

題目頁可以自行設計排版, 也可以利用現成之指令。若自行設計, 指令與文字內容可置於 `titlepage` 指令環境內:

```

\begin{titlepage}
...
\end{titlepage}

```

若利用現成指令，排版題目的指令為 `\title`，排版作者名字可使用 `\author`，日期指令為 `\date`，如前面圖 7.2 (頁 106) 之例子所示。各項指令都會自動選用適當大小的英數字體，但中文字體必須由 `ctex` 指令設定。本例以粗黑體字排版標題，大小則依內定值。

在 `\author` 指令之後，我們可使用 `\thanks` 指令排版致謝詞，其內容將以註解形式出現在題目頁下方，致謝詞之註解是以星號標示。如果你對於英文與數字字體之格式不滿意，也可以改變之。圖 7.2 例子中，若嫌內設值設定之英文字體太大，我們可以用 `\footnotesize` 指令縮小之，例如：

```
\date{\footnotesize 2002.1}
```

如果省略 `\date` 指令，`TeX` 將自動填入排版當天的日期。若不希望列出日期，可以使用 `\date{}` 指令。兩個大括號緊接在一起，表示日期指令內容空白。同樣的，如果我們下 `\author{}` 指令，作者名字將不出現。但如果完全省略 `\author` 指令，排版時將產生錯誤訊息。

題目、作者等指令輸入完畢後，還必須下 `\maketitle` 指令，指示以上為題目頁內容。這道指令通常是緊接在 `\begin{document}` 指令之後。如果使用 `book` 或 `report` 文件類別，題目、作者與日期將排版在單獨的一頁上，正文則從下一頁開始，請見圖 7.5 之例子。本例係採用 `report` 文件類別。

作者有兩人或兩人以上時，`\author` 指令內兩位作者名字可以並列，例如：

```
\author{ 吳慕凡。吳浩文}
```

事實上，這與單一作者的排版並無不同，唯一不同的是我們在兩位作者名字中間加上一小點。相反的，圖 7.5 的例子就稍為複雜一些，因為每一位作者各有其工作單位。下指令的方法是把第 2 位作者的名字及感謝詞置於第一位作者名字與謝詞之後，但兩位作者名字之間必須加入 `\and` 指令。我們以換行指令 `\\` 區隔作者名字及其服務單位。換行指令 `\\` 的作用是使服務單位移到下一行，並對齊作者名字。

前面圖 7.2 (頁 106) 的例子是以 `article` 文件類別排版。其中，`\thanks` 指令排版之致謝詞是以 * 號標示。在 `report` 與 `book` 文件類別下，致謝詞則以阿拉伯數字標示。圖 7.5 例子的第 2-3 行利用 `\fnsymbol` 指令將阿拉伯數字改為星號。標題頁排版完成之後，再以類似的指令將註解標示改回阿拉伯數字，見第 13-14

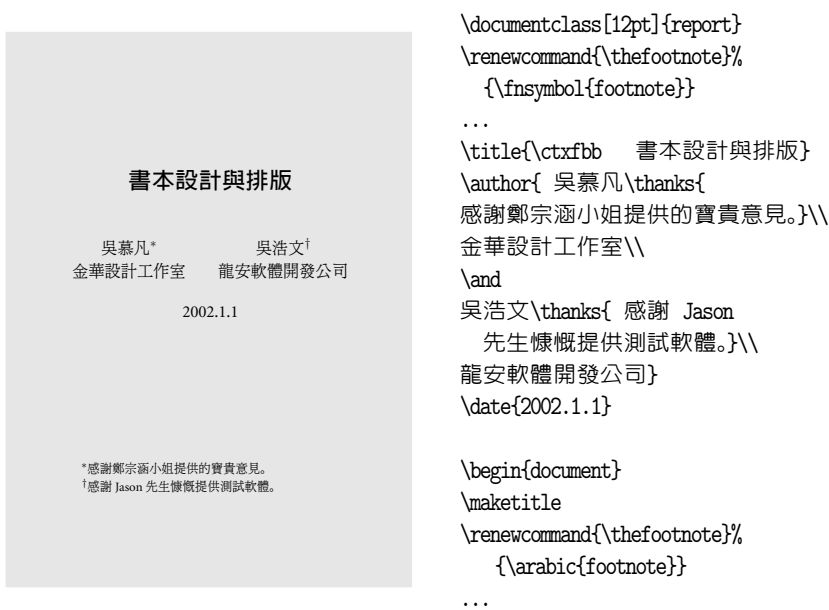


圖 7.5: 作者與感謝詞

兩行指令。有關於變更註解標示符號之方法，詳見 8.6 節 (頁 143) 之說明。

如果文章題目很長，一行排列不下，我們可以用換行指令 `\\` 拆成兩行或三行。譬如，經濟學名著《國富論》的英文原名甚長，輸入時可以拆成兩行：

```

\title{An Inquiry Into The Nature And\\
      Causes Of The Wealth of Nations}

```

排版之後，兩段文字都將居中排列。

如前所述，如果對於 \LaTeX 所編排之格式不滿意，我們可以自行設計題目頁之版面。此時，可以將排版題目、作者、名字等指令置於 `titlepage` 指令環境中。請注意，在此指令環境內 `\title` 與 `\author` 等指令皆不能使用，指令環境之後也不須下 `\maketitle` 指令。

7.4 摘要

文章正文之前可能排版摘要，摘要可以 `abstract` 指令環境編排。 \LaTeX 會在摘要文字之前加上 **Abstract** 英文字。如果是中文稿，我們須將此一英文字轉換為中

文。L^AT_EX 是以 `\abstractname` 指令設定摘要之標題文字。因此，只要重新定義其內容即可解決問題，辦法是在全文設定區加入下面指令：

```
\renewcommand{\abstractname}{\ctxfr 摘要}
```

此一指令重新設定摘要標題為中文「摘要」兩字，字體採用圓體字。

在 `article` 文件類別裡，摘要將排版於標題下面；在 `report` 文件類別裡，摘要自成一頁，而且不編上頁碼；`book` 文件類別裡則不能使用 `\abstract` 指令。

7.5 章節標題

一般文稿通常區分章節，章節的編排設計必須前後一致。L^AT_EX 提供現成的指令以編排章節；一般而言，這些指令已可滿足大多數人的需求。不過，中文與英文究竟不同。譬如，英文常使用“Chapter 1”作為第一章之編號，中文則使用「第 1 章」或「第一章」。如何編排適當的中文標題，對於 L^AT_EX 使用者而言是一個挑戰，主要原因是直接修改標題指令並不容易。幸運的是，我們有一些功能甚佳的巨集套件可資使用。

7.5.1 章節標題之層級

首先，我們介紹 L^AT_EX 排版章節標題之指令。使用這些指令排版時，章節標題上下會留出適當空白，小節以上之標題會自動編入號碼，標題內之英文/數字選用粗體字排版。文稿若設定排版頁眉/頁足，章節標題與編號會自動出現於頁眉或頁足。此外，若以 L^AT_EX 之指令排版目錄，章節標題/編號也會自動編入目錄內。L^AT_EX 之章節指令如下：

```
\part ( 部)          \chapter ( 章)
\section ( 節)       \subsection ( 小節)  \subsubsection ( 次小節)
\paragraph ( 段)    \subparagraph ( 小段)
```

章節標題指令有其層級結構，原則上先後順序不能倒置。例如，某一章之內可含有好幾個節；某一節之內又可分成幾個小節；其下又分次小節等等。

章節標題指令與文件類別關係密切。若使用 `book` 文件類別排版，以上各指令皆可使用；不過，`\part` 指令可跳過不用，直接從下一層級的 `\chapter` 指令

開始用起。相對而言，若是使用 `article` 文件類別排版短文，文稿內不能使用 `\chapter` 指令。但 `\part` 指令可有可無；若不使用，章節層級結構即由 `\section` 開始。

圖 7.2 (頁 106) 的例子中，各節標題是以 `\section` 指令編排，小節標題則以 `\subsection` 排版。文稿中遇有節與小節標題指令時，文稿將另起一行，自動編上號碼，選用稍大字體，再排出標題文字；正文內容將排於再下一行。節標題所選用之字體較小節字體大；小節標題字體又比次小節字體大一些。請注意，中文字體之大小與種類須以中文字體指令自行設定。若文稿甚長，每個章節標題都須輸入中文字體指令不僅不方便，也容易產生錯誤。7.5.4 節 (頁 119) 將說明如何定義整篇文稿的中文標題之字體。

在 `book` 文件類別中，若使用 `\chapter` 指令排版章標題，章編號與標題文字將靠左，並分上下兩行排出。下一層級的 `\section` 與 `\subsection` 指令所排版之節與小節標題，其格式與 `article` 文件類別類似；數字編號之後即接著標題文字，正文內容將從再下一行開始。`\subsection` 以下之標題皆不會自動編號，其中 `\subsubsection` 指令所排版之標題文字會再小一些，也是單獨成一行靠左編排。`\paragraph` 指令所排版之格式在專業排版中稱為 `running-head` 標題。標題文字與上一段之距離會稍大一些，標題使用粗體字，但大小與正文一樣。標題文字之後會留一點空白，其後直接排版正文。`\subparagraph` 之格式與 `\paragraph` 類似，但前者之標題會內縮 (`indent`)；後者之標題左邊則切齊版面左緣。

如果文稿一開始就跳過 `\section` 指令，直接使用 `\subsection` (小節)，文稿仍可編排，但第 1 小節的編碼將變成 **0.1**，第 2 小節變成 **0.2**，等等。因此，除非有特別理由，章節指令應按順序使用。次小節 `\subsubsection` 以下之標題不會自動編號，一方面是避免編號過於複雜，另一方面則是讓段落編排更有彈性。譬如，排版者在某一節或小節之內可以跳過 `\subsubsection` 指令，直接使用 `\paragraph` 指令以 `runing-head` 格式排版特定之段落。

以 \LaTeX 的指令排版中文時，面臨兩個問題。第一是中英文字體之搭配是否適宜，第二個問題是中英文章節編號之格式不同，以下進一步說明解決的方法。章節標題指令中，`\part` 指令適合於排版書籍或較長報告之標題。`\part` 指令之排版格式與 `\chapter` 類似。例如，若以下列指令排版標題：`\part{On Movies}`，版面上將先排出 **Part I**，下一行再以粗體字排版 **On Movies**。文稿內第 2 次出現 `\part` 指令時，版面上將出現 **Part II**，其下再排出標題文字。換言之， \LaTeX 自動

編上的序號是英文字的 **Part I** 與 **Part II**。如果是排版中文書，英文序號與中文標題可能並不搭配。使用節指令時， \TeX 也自動編上號碼，不過是以阿拉伯數字編碼，用於中文稿並無不妥。

如果要使用章節標題指令，但不要自動編號，應在章節指令之後加上 * 號，譬如 `\section*` 或 `\subsection*`。使用加 * 號之指令編排節標題，雖然不再編上號碼，但標題前後仍留出空白，標題之數字或英文選用較大字體，而且首段文字不內縮。使用 * 號節指令的一個問題是，節標題不會自動編入目錄與頁眉之中。如果節標題不想編號，但又希望編入目錄與頁眉中，須使用 `\addcontentsline` 指令，請參見圖 7.3 (頁 108) 之說明。

7.5.2 設定章節標題之字體

\TeX 之章節指令除了留空白與編號碼之外，並選用放大的粗體字。但是， \TeX 之字體指令無法改變中文字體。因此，我們必須自行下指令改變中文標題的字體。前面圖 7.2 (頁 106) 的例子以 `article` 文件類別編排，節與小節標題之中文字體是以中文字體巨集指令 `\ctxfdef` 定義，兩者都採用圓體字。但是，排版效果並不十分理想，原因是節與小節之編號數字字體是採用內定之英文羅馬字族粗體。如果中文標題使用圓體，較搭配之英文與數字應該是 `sans serif` 字族。我們如何改變標題之英文與數字字體呢？最簡單的方法是使用下一小節所介紹之 `titlesec` 巨集套件。不過，此處先介紹一般性之概念。

在 \TeX 中，章節之編號是由章節編號計數器 (counter) 自動調整。以節編號為例，其數值是由 `section` 計數器決定。文稿一開始，`section` 之值為 0。碰到第 1 個 `\section` 指令時，計數器值成為 1。碰到第 2 個 `\section` 指令時，計數器再加 1 變成 2。不過，`section` 計數器只是計算節之編號數字，實際之排版指令為 `\thesection`。要將節編號之字體改變為 `sans serif`，可在全文設定區加入指令：

```
\renewcommand{\thesection}{\textsf{\arabic{section}}}
```

如果小節編號也要作同樣改變呢？由以上的例子可知，第 1 小節之編號為 1.1，第 2 小節之編號為 1.2。因此，小節之編號用上兩個數字：節編號與小節編號。要改變小節編號之字體，我們不能只變更小節編號部分，必須同時調整兩者才能得到正確的結果：

```
\renewcommand{\thesection}{\textsf{\arabic{section}}}
\renewcommand{\thesubsection}%
{\thesection.\textsf{\arabic{subsection}}}
```

第 1 行指令之作用與上例完全相同，第 2 行指令定義小節數字編號之排版方式。我們先以 `\thesection` 指令排入節編號，其後緊接一英文句點，接著再排版小節編號。

根據以上原理，節編號可以作進一步的變化。L^AT_EX 的指令 `\alph` 可以將阿拉伯數字變成對應的小寫英文字母。因此，上面第 2 行指令若改為：

```
\renewcommand{\thesubsection}{\thesection.\alph{subsection}}
```

小節編號將變成 1.a, 1.b, ... 等。另外一個指令 `\Alph` 是將阿拉伯數字變成對應的大寫英文字母。如果第 2 行維持不變，但第 1 行指令之 `\arabic` 改為 `\Alph`，節編號將變成 A, B, C, ...；第 1 節之下的小節編號將變成 A.1, A.2, ... 等等。

L^AT_EX 改變數字排版格式之指令計有下列 5 個：

<code>\arabic</code>	阿拉伯數字，	<code>\alph</code>	小寫英文字母，
<code>\roman</code>	小寫羅馬字，	<code>\Alph</code>	大寫英文字母，
<code>\Roman</code>	大寫羅馬字。		

以上指令可以和表 7.4 改變頁碼字體之指令對照。

以上例子是以 `article` 文件類別為例。如果是 `book` 或 `report` 類別，編號是從章開始。因此，第 1 章第 1 節之編號為 1.1，第 2 節編號為 1.2；而第 1 章第 1 節第 1 小節之編號則為 1.1.1；餘此類推。

7.5.3 titlesec 巨集套件

要改變標題排版格式，最簡單的方法是使用現成的巨集套件。本節將介紹 Javier Bezos 所寫的 `titlesec` 巨集套件。此巨集套件提供簡易與進階兩種指令方式以變更章節標題。如果你只是要改變標題字體與大小，或者只是要改變標題之排版位置，簡易指令即可滿足需求。如果要進一步更改標題設計，則須使用進階指令。本節介紹簡易指令之使用方法，進階指令請見 17.2 節（頁 326）之說明。

欲使用簡易指令，僅須在引用巨集套件時直接加入控制格式之選項即可，表 7.3 列出簡易指令之選項。譬如，若標題文字要改為居中編排，僅須在全文設定區

表 7.3: titlesec 巨集套件簡易指令

<code>\usepackage[options]{titlesec}</code>
□ 字體: <code>rm sf tt md bf up it sl sc</code>
□ 字級: <code>big medium small tiny</code>
□ 標題位置: <code>center raggedright raggedleft</code>
□ 間距: <code>compact</code>

加入下列兩行指令:

```
\usepackage[center]{titlesec}
\renewcommand{\chaptername}{}
```

第 1 行指令中 `center` 選項之作用是將章節標題全部改變成居中排版。若文稿是採用 `book` 文件類別, 使用 `\chapter` 指令編排章標題時, 第 1 章標題文字之上方將自動加入 **Chapter 1**, 第 2 章自動加入 **Chapter 2** 等等。

在 \LaTeX 中, `\chaptername` 指令代表 **Chapter** 英文字, 本例之第 2 行指令即將 `\chaptername` 設為空白, 因此章標題之 **Chapter** 一字即不出現。排版之後, 第 1 章標題分上下兩行, 第 1 行為 **1**, 第 2 行為標題文字。以上指令雖可去掉 **Chapter** 一字, 但版面上方卻會顯得空盪盪。必要時, 我們可以使用 `\vspace*` 指令將標題往上移一些。

為方便參考, 我們將 `titlesec` 巨集套件之簡易指令全部列於表 7.3。控制標題位置之選項除了 `center` 之外, 尚有 `raggedleft` (標題靠右) 與 `raggedright` (標題靠左)。若選項有兩個以上, 須以逗號分隔。字體/字級之指令選項僅對英數字有效; 中文字體與字級之選擇須以中文字體指令另行設定。 \LaTeX 章節標題之英數字是以粗黑體 **text** 排版。如果中文標題是選用圓體或粗黑體, 則英文或數字改用 `sans serif` 字體搭配較適當。此時, 可使用下列指令:

```
\usepackage[sf,small]{titlesec}
```

其中, `small` 選項指示使用較小一點的字體; 內設值為 `big`。此外, 選項中若加入 `compact`, 則標題文字與上下文之間距會縮小一些。

傳統洋文書常在章節數字編號前後加上裝飾符號, `titlesec` 巨集套件所提供之 `\titlelabel` 指令可用以排版裝飾符號。 \LaTeX 是以 `\thetitle` 指令代表整個

章節之數字編號，若要在章節編號之前加上 § 符號，僅須在全文設定區加入底下一行指令即可：

```
\titlelabel{\S\ \thetitle\quad}
```

此行指令中，\S 指令的作用是加入裝飾符號 § 於 \thetitle 之前，\quad 指令是用以加大章節編號與標題文字之間距。排版後，某書 7.5 節之標題編號將變成：§ 7.5。以上指令雖然方便，但它會更動文稿中全部章節之標題。除非文稿結構單純，否則不一定適用。

7.5.4 中文標題之字體

使用 \TeX 編排章節時，標題內之英數文字會自動選用較大的粗黑字體。對某些人來說，章節標題不須另行設計是 \TeX 排版系統的優點。但對於想自行設計章節標題的人來說，固定格式反而是缺點。上一小節所介紹的 `titlesec` 巨集套件讓更改文稿標題之設計變得很容易；不過，其中仍然有一個不方便之處，那就是中文標題須在每一個章節標題處下指令更改。

爲了讓使用者能一舉設定章節標題之中文字體， c\TeX 提供一簡單的巨集指令：`\ctxfdef`。舉例言之，若文稿內使用 `\section` 與 `\subsection` 指令排版節與小節標題。假設節與小節標題之中文字都選用粗黑體，則在全文設定區加入下列兩行指令即可：

```
\ctxfdef{\section}[\ctxff]{\ctxfbb}
\ctxfdef{\subsection}[\ctxff]{\ctxfbb}
```

第 1 行指令設定節標題之中文字以粗黑體排版，中括號內之 `\ctxff` 指令設定節標題文字排版於目錄及頁眉或頁足處之字體。若不排版頁眉或目錄，中括號選項可以省略。同理，第 2 行設定小節標題之中文字體。

經過以上設定，排版本節標題之指令可簡化爲：

```
\section[ 章節標題]{章節標題}
```

請注意，中文標題須鍵入兩次，中括號內者是用於排版頁足與目錄，大括號內則用於排版實際標題。定義 `\ctxfdef` 指令之後，`\section` 指令大括號內不須再加

入中文字體指令。如果某一節是以 `\section*` 指令排版，則 `\ctxfdef` 所定義之中文字體仍然有效。

若全文設定區已以 `\ctxfdef` 指令設定中文字體，而文稿內某特定章節欲改用不同字體，可在該章節指令處再加入中文字體指令。例如，若 7.5 節 (頁 114) 之標題欲改用楷體，指令為：

```
\section[ 章節標題]{\ctxfk 章節標題}
```

7.6 頁碼與頁面格式

任何文稿至少都會排版頁碼，長篇文稿還會在頁眉或頁足排版章節標題。頁眉或頁足之排版， \LaTeX 稱為頁面格式 (page style)。頁碼通常是排版於頁眉或頁足，實際上是頁面格式的一部分。本節將介紹排版頁面格式之指令。在某些情況下， \LaTeX 指令在排版頁眉之中文標題時會出現亂碼，此時，我們須使用 Piet van Oostrom 所寫的 `fancyhdr` 巨集套件，請見 17.4 節 (頁 338) 之介紹。

欲設定頁面格式，我們在全文設定區加入指令：

```
\pagestyle{plain}
```

其中，`plain` 為四個可能的選項之一。可能的選項如下：

<code>empty</code>	頁眉與頁足全部空白，
<code>plain</code>	頁碼置於頁足正中央，頁眉空白，
<code>headings</code>	頁碼與章節標題等資訊自動排版於頁眉，頁足空白，
<code>myheadings</code>	與 <code>headings</code> 類似，但章節標題的排版可以自行控制。

若不加任何指令， \LaTeX 將選用 `plain` 格式，頁碼自動加於每頁下方，頁眉為空白。這是最常見之頁面格式。

7.6.1 頁碼

依專業排版之規範，書籍每一章標題頁通常不排版頁碼。`\pagestyle` 若採 `plain` 格式， \LaTeX 會自動取消章標題頁的頁碼。但是，若以 `article` 文件類別排版數頁之傳單，而且每頁上不希望有頁碼出現，我們須加上指令消除。

表 7.4: 頁碼數字之格式

arabic	阿拉伯數字 (內定),
roman	小寫羅馬字, 如 i, ii, ...,
Roman	大寫羅馬字, 如 I, II, ...,
alph	小寫英文字母, 如 a, b, ...,
Alph	大寫英文字母, 如 A, B, ...。

要更改特定頁之頁碼, 須使用 `\thispagestyle` 指令。相反的, 若是要改變文稿中的每一頁, 應使用 `\pagestyle` 指令。譬如, 在全文設定區加入下列指令:

```
\pagestyle{empty}
```

文稿每一頁的頁碼都不出現。反之, 在特定點加上 `\thispagestyle{empty}`, 該頁之頁碼即不出現。

在正常的情况下, 頁碼是從 1 開始編排。如果頁碼要重新定為其他數字, 例如 20, 須透過頁碼計數器 (page counter)。計數器是 \TeX 記錄章節編號、頁碼數目等之變數。頁碼數目是記錄於 page 計數器內; 欲改變頁碼, 可使用 `\setcounter` 指令:

```
\setcounter{page}{20}
```

以上指令將本頁頁碼改為 20。若某書共有 9 章, 排版時每一章分開編排。假設第 2 章是從第 20 頁開始, 我們即可在第 2 章檔案前端加入上述指令。

頁碼之數字是以阿拉伯數字編排, 但也可以改成羅馬數字甚至英文字母。舉例言之, 欲改用小寫羅馬數字編排頁碼, 請在全文設定區輸入下列一行指令:

```
\pagenumbering{roman}
```

其他可用之選項如表 7.4 所示。`\pagenumbering` 指令除了改變頁碼數字格式之外, 頁碼計數也會由 1 重新開始。請注意, 小寫字母之指令為 `alph`, 而非 `alpha`。這可能是為了與排版希臘字母 α 之指令 `\alpha` 有所區別。同理, 設定大寫字母之指令為 `Alph`。

某些洋文書的目錄或序文是以小寫羅馬字排頁碼, 正文之頁碼才使用阿拉伯數字。若欲追隨此項傳統, 可使用 `\frontmatter` 與 `\mainmatter` 指令。若在標

題頁內容之前加上 `\frontmatter` 指令，頁碼數字將改以羅馬數字編排。開始排版正文之前應下 `\mainmatter` 指令，頁碼將改以阿拉伯數字編排，並重新從 1 起算。`\frontmatter` 指令除了改變頁碼數字格式之外，還有其他效果。

排版書籍時，`\chapter` 指令可用以編排每章標題文字， \TeX 會自動加入章編號。因此，文稿中第一道 `\chapter` 指令即為本書第 1 章。但是，在 `\frontmatter` 與 `\mainmatter` 指令之間所下之 `\chapter` 指令， \TeX 仍以一般章標題格式編排，但不加入數字編號。此項特別設定讓我們可以利用 `\chapter` 指令排版序言或目錄。不過，文稿中一旦碰上 `\mainmatter` 指令，其後之 `\chapter` 即恢復自動編碼功能。書籍末端若欲排版索引或參考文獻，可先加入 `\backmatter` 指令，其後之 `\chapter` 指令也不自動編號。

7.6.2 換頁

排版時， \TeX 會自動選擇適當的地方換頁。但是，若希望把某一段文字排在同一頁裡，而 \TeX 所選擇換頁的地方剛好是在這一段文字中間，我們可以在此段文字之前下換頁指令 `\newpage`，其下的整段文字將移至下一頁。我們也可以使用 `\clearpage` 指令，兩者的差別在於後者除了換頁之外，還會把尚未排出的圖表列印於本頁之後。若是使用 `\newpage` 指令，圖表將移於文稿最末端。

如果 `\newpage` 指令恰出現於一頁之頂端，該指令變成無效，亦即不會再空出一頁。因此，如果一定要空出一頁，應使用下列指令：

```
\mbox{} \newpage
```

其中，`\mbox{}` 創造出一虛擬字元，因此跳頁指令即可產生效果。

文稿若要排版成兩欄格式，我們可以在文件類別指令中加入 `twocolumn` 選項。在兩欄式版面中，換頁應使用 `\cleardoublepage` 指令。此一指令的功能與單欄版面中之 `\clearpage` 指令類似，但下接之文字會跳至單數頁開始編排。換言之，如果本頁為單數頁，下一頁（雙數）將為空白，文字會出現於再下一頁中。

排版書籍時，是否跳頁常依本頁是單頁或雙頁而定。譬如，若每一章標題是排於單數頁，而上一章結束於單數頁，則下一頁（雙數頁）須留為空白。反之，若上一章是結束於雙數頁，本章恰好接著由單數頁開始編排，不須跳頁。將下列一行指令加於每一章之前，即可出現以上的效果：

```
\ifodd\count0 \else \mbox{}\clearpage \fi
```

以上的 TeX 指令中, `\ifodd` 是判斷數字是否為奇數; `\count0` 則記錄頁碼。因此以上指令之意義如下: 若本頁頁碼為奇數, 不作任何動作; 反之, 若為偶數, 則強迫跳一頁。指令最後之 `\fi` 代表定義結束。

以上指令解決了先判斷單雙頁再決定是否跳頁的問題。不過, 跳空的那一頁雖然空白, 版面正下方仍然會排出頁眉/頁足。如果跳空的那一頁希望全頁空白, 上面之指令須修改如下:

```
\clearpage
\ifodd\count0 \else \thispagestyle{empty}\mbox{}\clearpage \fi
```

定義內 `\thispagestyle{empty}` 指令之功能是設定本頁之頁眉/頁足皆為空白, 頁碼自然也不會出現。

以上是以跳頁方法避免某段文字拆散於兩頁之中。除此之外, 我們也可以將 `\samepage` 指令加在文字段落之後。TeX 會試著把整段文字全部擠入本頁。頁面調整指令應在排版最後階段再加入。否則, 文稿一經修改, 又須重新調整, 徒然浪費時間。

7.6.3 頁眉與頁足

除了頁碼之外, 排版書籍或長篇文稿時, 通常會在頁眉或頁足排版章節標題。在 `\pagestyle` 中若使用 `headings` 或 `myheadings` 選項, 我們即可在頁眉排出章節標題。這些文字可能是內定值, 也可以重新定義。不過, 這兩個選項在中文稿會出現問題, 頁眉之中文標題會出現亂碼。要正確排版頁眉之章節標題, 我們可以使用 `fancyhdr` 巨集套件, 請見 17.4 節 (頁 338) 之說明。

以下之介紹僅適用於英文稿。我們首先說明 TeX 之標準設定。在 `book` 文件類別下, 文稿之左右頁對稱於書脊。左頁 (偶數頁) 書眉右邊將排版本章標題; 左邊則排版頁碼。相對的, 右頁 (奇數頁) 書眉左邊將排版本節標題; 右邊則排版頁碼。若採 `article` 文件類別排版, 左右頁之頁面格式相同, 頁眉之排版也不同於上面所述。不過, 如果採用 `article` 文件類別時加上 `twoside` 選項, 則書眉之格式與 `book` 文件類別相同。

相反的, 若以 `book` 文件類別但加上 `oneside` 選項, 則左右頁之格式相同。

換言之，這是 `article` 文件類別的標準格式。頁眉右邊將排版頁碼，左邊則排版節標題。爲了要排版頁眉， \LaTeX 將每一頁之章節標題記錄於 `\leftmark` 與 `\rightmark` 指令內。在 `book` 文件類別下，前一指令記錄章標題；後一指令記錄單數頁之資訊。在 `article` 文件類別下，`\leftmark` 記錄節標題，`\rightmark` 則記錄小節標題。當頁碼增加，由第 1 節變爲第 2 節，或由第 3 章變成第 4 章時，以上兩項指令所記錄之內容也隨著更新。

要變更頁眉之排版，我們可使用下列指令：

```
\markboth{ 左頁頁眉}{右頁頁眉}
```

```
\markright{ 右頁頁眉}
```

`\markboth` 指令用於設定左右頁版面不同之頁眉。譬如，我們正在排版某書之索引，而且文稿區分左右頁版面 (`book` 文件類別，或者 `article` 文件類別，但加上 `twoside` 選項)，若左右頁之頁眉都要出現「索引」兩字，應在開始編排索引處下指令：

```
\markboth{ 索引}{索引}
```

相對的，`\markright` 指令是用於左右頁相同版面之文稿。譬如，以 `article` 文件類別排版之論文，若希望在頁眉排出節標題可在全文設定區使用下列指令：

```
\pagestyle{myheadings}
```

```
\markright{\rightmark}
```

7.7 目錄

\LaTeX 能自動記錄章節與圖表標題之文字內容及其頁碼，因此排版目錄甚爲簡單。欲自動排版目錄，章節標題須使用 `\chapter`，`\section` 等指令編排，圖表標題則須以 `\caption` 指令編排，否則 \LaTeX 無法取得所需資訊。

根據 7.5 節之說明，章節指令有好幾層。依內定值，`book` 文件類別所排版文稿之目錄將排版到 `\subsection` 層級，短文 `article` 文件類別之目錄則排版到次小節。若增加或減少目錄之層級內容，須改變 `tocdepth` 計數器之值。譬如，於全文設定區加入下列指令：

```
\setcounter{tocdepth}{3}
```

則目錄將編排到 `\subsubsection` 層級。

章節標題若是以加 * 格式之指令編排, 如 `\section*`, 其內容不會自動排版於目錄內。欲將其內容自動編入目錄, 可使用 `\addcontentsline` 指令:

```
\addcontentsline{toc}{ sec-name}{text}
```

請見 17.2.4 節之說明。此外, 如果要在目錄內特定地方加入排版指令或文字, 可使用以下指令:

```
\addtocontents{toc}{ text}
```

目錄頁通常是排版於正文與序言之前。在文稿中選定目錄頁所要出現之位置, 鍵入 `\tableofcontents` 指令, 執行 latex 編排文稿兩次, 章節目錄內容即出現於指定位置。目錄首頁將出現 **Contents** 標題, 此英文標題可以變更為中文字。例如, 在全文設定區加入下列指令:

```
\renewcommand{\contentsname}{\ctxfbb 目錄}
```

標題將以粗黑體排出。

目錄是以內定之格式編排, 若不滿意其格式, 也可以自行設計。欲自行設計目錄排版格式, 最好是使用 `titletoc` 巨集套件, 請見 17.3 節 (頁 335) 之說明。另外一個辦法是利用 \TeX 所產生之目錄原始資料, 再自行排版。假設文稿檔名為 `test.ctx`, 以 latex 編排兩次或三次之後, 工作檔案夾內會出現 `test.toc` 輔助檔案, 其內容即為各章節之標題與頁碼。若為中文稿件, 章節標題之中文字是經過 `cwTeX` 程式轉換後的格式, 而非原始輸入之中文字。但我們可以利用 `cwTeX` 提供的工具程式 `tex2xtc` 轉回中文字碼:

```
c:\xtemp>tex2xtc test.toc
```

執行指令之後, 將產生 `test.xtc` 檔案, 其內含有中文章節標題與頁碼。將此檔案修改為理想的格式後, 即可引入原檔案內進一步排版。

若要排版圖標題之目錄, 文稿中須下 `\listoffigures` 指令; 同理, 排版表格標題目錄之指令為 `\listoftables`。排版之後, 工作檔案夾內分別產生 `test.lof` 與 `test.lot` 兩個檔案。圖標題目錄頁上方會排出 **List of Figures**, 表目錄頁上方則為 **List of Tables**。若欲改為中文標題, 請做照上面的作法, 使用 `\renewcommand`

指令重新定義 `\listfigurename` 之內容與 `\listtablename` 指令之內容。或者，我們也可以利用 `tex2xtc` 程式，將 `.lof` 與 `.lot` 檔案轉換為中文檔，再進一步編輯排版。

7.8 附錄

\TeX 提供 `\appendix` 指令以排版附錄。根據 \TeX 的設計，附錄是指在文稿末端特闢一塊排版區域，在此區域內我們仍使用章節指令，但章節之編號改變。舉例來說，在 `book` 與 `report` 文件類別中，若文稿末端加上 `\appendix` 指令，它產生下列作用：

1. 正文結束，另起新的一頁。
2. 附錄第 1 章的標題仍以 `\chapter` 指令編排。例如，`\chapter{ 資料來源}` 指令將使標題排為兩行，第一行為 “Appendix A”，第二行為 “資料來源”。其下之 `\section` 指令之編號變成 A.1。若附錄內有第 2 個 `\chapter` 指令，其章編號將為 B。

若是中文稿件，我們可以使用以下指令將 **Appendix** 一字改為「附錄」或其它中文標題：

```
\renewcommand{\appendixname}{\ctxfbb 附錄}
```

以上是 `book` 與 `report` 文件類別的情況。若是使用 `article` 文件類別排版，則 `\appendix` 指令並不產生 “Appendix” 一字，因此也無 `\appendixname` 指令。文稿中下 `\section{ 資料來源}` 指令時，排版結果為「A. 資料來源」。不過，如果文稿引入 Peter Wilson 之 `appendix` 巨集套件，則 `article` 文件類別下也會有 `\appendixname` 之定義。

\TeX 的附錄指令在某些情況下並不適用。譬如，若某書分數章，而每一章都要排版附錄，則 `\appendix` 指令無法處理。不過，根據以上說明，`\appendix` 指令基本上僅是改變了章節編號方法。因此，要排版較複雜的附錄，我們不妨自行變更版面設計。有關於版面設計之巨集套件，請見第 17 章之介紹。

8 段落編排

上一章說明文稿結構與章節設計，本章將說明排版局部段落之指令，如迷你版面、註解、列舉項目等等。從排版的角度來看，版面上最基本的元素是單字。由單字組成句子、句子再組成段落、眾多的段落則構成章節。因此，段落編排是很重要的一部分。

要編排普通的文字段落很簡單，我們在文稿中留出一行空白，表示上段落結束，以下為新段落開始。L^AT_EX 會將此段文字排得整整齊齊。但有些段落需要特別的設計，譬如，引述一段話、排版詩歌、條列說明等。這些特別的段落都是利用指令環境來編排。

除此之外，本章也將介紹照列原文 `verbatim` 指令環境，以及交叉引述 (cross-reference) 的指令。排版長篇文章時，各章節之間有時候要相互引述。譬如，第 5 章內文可能引述 3.2 節的文字或圖 4.1。L^AT_EX 所提供的 `\label`，`\ref`，與 `\pageref` 三道指令即作此用途，我們將於 8.8 節介紹。最後，我們也簡單說明如何排版多欄式版面。

8.1 段落格式

一般文稿在新段落開始處會內縮一小段距離，在英文中這稱為 `indent`。此一工作可以由 `\indent` 指令來完成。輸入文稿時，空一行 (或多行) 即表示要起新段落。當 L^AT_EX 碰到一空白行時即結束本段，同時在下一段開頭自動加上 `\indent` 指令。因此，除非有特殊的情況，我們不須再下 `\indent` 指令。如果不希望句子內縮，我們可以在段落開始之處使用 `\noindent` 指令。

若以 12 點之字體排版，則段落開頭內縮之距離約為 0.6 公分。不過，此一距離可以自行選擇。譬如，要將段落開頭內縮之距離拉長為 0.8 公分，只要在全文設定區加入下列指令即可：

```
\parindent=0.8cm
```

上面說明，若要起新段落，輸入文稿時必須留一空行。另外一個辦法是在上一段落結束處加上 `\par` 指令，其作用和留空行完全相同。另外，上一段落最末一行與本段落第一行間的行距和一般的行距相同，並不特別加大。如果要加大段落間的行距，可以使用 `\parskip` 指令。譬如，若在全文設定區加上以下的指令：

```
\parskip=20pt
```

段落間的行距將改成 20 點。

段落內若要新起一行，可以將換行指令 `\\` 加於本行之末；下一行之文字將齊頭編排，不會內縮。本行與下一行之間的行距若要比正常行距加大 0.5 公分，可以加上選項：`\\[.5cm]`。如果行距要縮小，則選項應為負值，例如 `\\[-3pt]` 可以把行距減小 3pt。另一個對應的指令是 `*`。這個指令的作用也是換行，不過它可以禁止 \TeX 在指令處換新頁。`\newline` 指令與 `\\` 功能完全相同，但前者不能加上變更行距之選項。

8.1.1 居中與靠邊

排版時， \TeX 會儘可能讓每一行文字之右沿切齊。在西文排版中，這稱為 *justification*。但是，我們也可以取消右沿切齊之設定，讓每一行文字居中、靠左或靠右編排。首先，若在文稿某處輸入 `\centering` 指令，則底下的每一行文字將居中編排。若是普通的文字段落，則每一行之左右沿都不會上下對齊。

若要左沿上下對齊，右沿不切齊，可使用 `\raggedright`。反之，若要右沿上下對齊，左沿不切齊，則使用 `\raggedleft` 指令，請見底下例子。

標題居中	<code>\centering</code>	標題居中\\
可靠左	<code>\raggedright</code>	可靠左\\
或靠右	<code>\raggedleft</code>	或靠右

本例子之前兩行文字末端都加入換行指令，故排版成 3 行。排版兩欄式版面時，因為每一行文字較短，一般常採用右沿不切齊的格式排版，此時僅須在文稿前端加入 `\raggedright` 指令即可。

以上 3 個指令會影響指令底下全部的文字段落之編排。如果是要將某幾段文字或圖表居中或靠邊排版，則使用指令環境較方便。要將某段落居中排版，可使

用 `center` 指令環境, 靠邊則使用 `flushright` 或 `flushleft` 指令環境。底下以鄭愁予的詩簡單說明指令之用法。

```
我打江南走過,
那等在季節裡的容顏...
東風不來,...
```

```
\begin{center}
我打江南走過,\\
那等在季節裡的容顏 ...\\
東風不來, ...
\end{center}
```

本例中, `center` 指令環境內全部為文字, 但其中也可以含有其他的指令環境。第 10 章將說明排版表格之指令, 若要把表格排版於版面中央, 只要將整個表格之指令與文字置於 `center` 指令環境內即可。

如果要靠左排版, 應使用 `flushleft` 指令環境:

```
我打江南走過,
那等在季節裡的容顏...
東風不來,...
```

```
\begin{flushleft}
我打江南走過,\\
那等在季節裡的容顏 ...\\
東風不來, ...
\end{flushleft}
```

若要靠右編排, 則使用 `flushright` 指令環境。

以上所說明之居中排版是指將段落安排於橫向的中間位置。如果是要將某一段落安排於縱向的中間位置, 我們可以使用 `\vfill` 指令。首先, 在上一段落結束加上 `\newpage` 指令, 底下之文字段落將另起新的一頁。然後, 在段落文字之前後各加上 `\vfill` 指令, 即可產生縱向居中之段落。`\vfill` 指令的作用是将文稿段落往上頂或往下擠。

舉例來說, 若文稿最後一頁的文字並未填滿整頁, 而我們希望在該頁底部的左方記錄文稿檔名及日期, 一個簡單的方法是在 `\end{document}` 之前一行輸入底下的指令:

```
\par\vfill\noindent \jobname.ctx (\today)
```

以上指令中, `\par` 指令用於結束上一段落, `\vfill\noindent` 指令則把檔名及當天日期往下擠壓至最底端, 且靠左編排。`\jobname` 是 T_EX 指令, 代表排版文稿的主檔名, 我們須在其後自行加上延伸檔名 `.ctx`。

8.2 調整間距

排版時，我們經常須控制單字、句子或表格之位置，或者段落間之距離。上面已說明我們可以使用 `\parskip` 指令控制段落之間距；本節將進一步介紹調整段落內間距之指令。

8.2.1 插入空白

\TeX 有三個現成的垂直空白指令：`\bigskip`，`\medskip`，與 `\smallskip`。其中，`\smallskip` 所產生之間距最小，`\medskip` 所產生之間距為 `\smallskip` 的兩倍；`\bigskip` 所產生之間距又為 `\medskip` 的兩倍。事實上，`\smallskip` 指令相當於是：`\vspace{\smallskipamount}`，其中 `\smallskipamount` 是一個內定之數值。在正常情況下，其值為 3pt。我們之所以說「在正常情況下」，原因是此數值事實上是在 2pt 到 4pt 之間伸縮。換言之，其伸縮彈性為 1pt。當 \TeX 碰到 `\smallskip` 指令時，它會先考慮整個版面之空間，情況許可的話，它將空出 3pt 之垂直空白。若加入 3pt 之垂直空白會使版面變成太擁擠，空白可能只有 2.4pt。反之，若 3pt 之垂直空白使版面顯得太空盪，則空白可能加大到 3.8pt。

若要自行設定間距大小，可直接使用 `\vspace` 或 `\vspace*` 指令。前一項指令如果下在本頁末端，或者新一頁之起頭處，指令皆無效。相對的，後一指令不管是下在什麼地方，都產生垂直空白。要注意的是，垂直空白指令若下在段落中間，則空白是在本行之後才出現。我們在本段結尾加上 `\vspace*{.5cm}` 指令，因此段落間的空白比正常間距多了 0.5 公分。

對應垂直空白指令， \TeX 亦提供幾個現成的水平空白指令，較常用的是 `\quad` 與 `\qquad`。前者所產生之水平空白恰等於正文字體之點數。正文若使用 10pt 字體，則 `\quad` 產生 10pt 水平空白；`\qquad` 則產生兩倍之距離。

若要直接控制間距，可直接使用 `\hspace` 與 `\hspace*` 指令。譬如：

距離	1 公分	距離 <code>\hspace{1cm}</code>	1 公分 \\
距離	1 公分	距離 <code>\hspace{1cm}</code>	1 公分 \\
距離	1 公分	距離 <code>\hspace{1cm}</code>	1 公分 \\
後退 0.5 公分		<code>\hspace{-0.5cm}</code>	後退 0.5 公分

若距離值為負數, 文句將左移。此例中, 第4句開頭 `\hspace{-.5cm}` 指令讓句子左移0.5公分。

仔細比較例子的前三行, 可發現句子中間空白的大小和 `\hspace` 指令前後是否留有空白有關。如果指令之前留一空白, 版面上所產生的空白為1公分加上空白鍵之距離。如果後面也留空白, 空白為1公分加上兩個空白鍵之距離。和 `\hspace` 指令相近的是 `\hspace*`。前一指令如果恰好出現在一行開頭處, 則指令無效, 本行文字開頭並不會右移。反之, 後一指令不論是在什麼地方出現, 都將產生水平空白。

有時候, 我們要把一行文字拆開為兩段, 並儘量往左右兩邊靠, 或者是要將垂直空白插入一頁當中; 此時可應用 `\hfill` 或 `\vfill` 指令。我們可將 `\hfill` 指令想像成是一個有力的彈簧, 如果加在一行中央, 它會將前後的文字往左右兩邊推擠。例如, 若輸入下列指令:

```
1996 年 7 月\hfill 台灣大學\
1996 年 7 月\hfill 台灣經濟史\hfill 台灣大學\
```

排版結果為:

1996 年 7 月		台灣大學
1996 年 7 月	台灣經濟史	台灣大學

本例中, `\hfill` 指令事實上等於是 `\hspace{\fill}`。 `\hspace` 指令是留出水平空白, 但我們設定留出的空白是一個會往外伸張的彈簧 `\fill`。第一行指令中, 彈簧左右兩邊的文字都被往外推。第2行指令裡有兩個彈簧, 因此「台灣經濟史」5個字就從兩邊往中間擠壓。上一節曾說明居中排版指令: `\centering`, 由以上例子可知, 兩個 `\hfill` 指令也可產生同樣效果。

若要將文字靠左排版, 可以利用下列指令:

```
\noindent ... \hfill\par ,
```

其中 ... 代表排版文字。以上指令相當於上一節所介紹之 `\raggedright`。類似的, 靠右排版 `\raggedleft` 的對應指令是: `\hfill ...\par`。

8.2.2 填入細點或直線

在一行文字中加入水平方向之空白,可使用 `\hfill` 指令;相對的, `\dotfill` 指令可以在加入的空白中填入細點。如果要在產生的空白中畫出一水平線,應使用 `\hrulefill` 指令。以上的指令中, `\dotfill` 的應用之一是排版目錄。例如,下列的指令:

- 1. 前言 `\dotfill 1\`
- 2. 例子 `\dotfill 5\`
- 3. 排版方法 `\dotfill 17`

排版結果是:

1. 前言		1
2. 例子		5
3. 排版方法		17

若使用 `\hrulefill` ,點線將變成實線,請見以下例子:

名字: `\hrulefill`
 職稱: `\hrulefill\hspace{1cm}` 公司: `\hrulefill`

本例中,「職稱」之後有兩個 `\hrulefill` 指令,其中夾著 1 公分的空白及「公司」兩個中文字。排版結果為:

名字: _____
 職稱: _____ 公司: _____

8.3 引文與詩詞

\TeX 雖然是以排版數學文稿著稱,但也有很多用它來排版文學作品。 \LaTeX 提供 `verse` 指令環境,可用於排版詩詞。欲徵引其他文章段落,可使用 `quote` 或者 `quotation` 指令環境。

8.3.1 quote 指令環境

文稿中若引用他人的文句, 可以使用 `quote` 與 `quotation` 指令環境。前者用於排版較簡短的引文, 後者適用於排版長篇段落之引文。若是引文是中文稿, 我們可以利用 `\ctxfdef` 指令定義特別的字體。請見以下例子:

在國慶文告的最後, 偉大的領袖一定會說:

讓我們一起高呼, 反共必勝! 建
國必成! 中華民國萬歲! 蔣總統
萬歲! 萬歲! 萬萬歲!

全國民眾歡欣鼓舞不已。

```
\ctxfdef{quote}{\ctxfk}
```

...

在國慶文告的最後,
偉大的領袖一定會說:

```
\begin{quote}
```

讓我們一起高呼, 反共必勝!
建國必成! 中華民國萬歲!
蔣總統萬歲! 萬歲!
萬萬歲!

```
\end{quote}
```

全國民眾歡欣鼓舞不已。

指令環境 `quote` 通常用於簡短的引文。如果引文甚長, 分開成好幾段, 則使用 `quotation` 指令環境較為方便。在此環境之下, 各文字段落也是以空白行區隔。排版之後, 引文的每一新段落開始將內縮一些。相對的, 在 `quote` 指令環境下, 新起之段落行首不會內縮。

8.3.2 verse 指令環境

若要排版詩詞歌謠, 可以使用 `verse` 指令環境。底下例子取自呂泉生作曲的〈阮那打開心裡的門窗〉:

〈阮那打開心裡的門窗〉是由呂泉生作曲, 描述對故鄉 ...

阮那打開心裡的門,
就會看見故鄉的田園。
雖然路途千里遠,
總是 ...

〈阮那打開心裡的門窗〉

是由呂泉生作曲, 描述對故鄉 ...

```
\begin{verse}
```

阮那打開心裡的門,\\
就會看見故鄉的田園。\\
雖然路途千里遠,\\
總是 ...

```
\end{verse}
```

我們在每句之後加上 `\\` 指令以確定分行。一首詩歌若區分數個段落, 段落間應以空行分開。排版之後, 詩詞每一行會自動內縮一點。

8.4 條列指令環境

文稿中常常出現條列式文字，或作列舉，或作摘要式說明。L^AT_EX 提供三種條列指令環境：`itemize`，`enumerate`，與 `description`。每一種條列指令環境各有特定格式。必要時，我們還可以自行定義格式。不過，以下僅說明三種現成的指令環境之用法。

條列指令環境內還可以再使用另一層的條列指令環境，最多可以使用 6 層。不管是那一種指令環境，每一列舉項都是以 `\item` 指令起頭。排版之後，每一條列項句子之前會有一文字或符號標籤 (label)。在 `itemize` 與 `enumerate` 指令環境下，條列項之標籤有內定之符號，但也可以自行設定。`description` 指令環境之標籤則由使用者自行輸入。

若使用 `itemize` 指令環境，L^AT_EX 會先在各條文之前加上一個圓點，其後再排版條文內容。若 `itemize` 環境下還有一層 `itemize` 指令環境，則內層之各項條文將以短線作為標籤。內層若使用 `enumerate` 指令環境，條文之標籤是阿拉伯數字。再下一層的 `enumerate` 環境中，標籤是小寫英文字母 (a), (b), (c) 等等。

圖 8.1 之例子先以 `itemize` 指令環境起頭，其下是 `enumerate` 指令環境。在 `enumerate` 環境之下，含有兩個次一層的 `enumerate` 環境。在以上三個 `enumerate` 指令環境之後，我們使用另一個 `itemize` 指令環境。指令環境是以 `\begin{...}` 與 `\end{...}` 的形式成對出現。當同時使用多個條列指令環境時，須小心起頭與結尾的配對指令不可搞錯。排版時，每一列舉項之上下各會留出較大的空白。如果要調整空白的大小，必須另外下控制指令。底下會有進一步的說明。

使用條列指令環境時，L^AT_EX 自動在列舉條文之前加上內定之標籤符號。但我們也可以自行選用符號或文字。例如，圖 8.1 中第一個 `\item` 若改為 `\item[A]`，排版之後標籤將為 **A**。反之，若改為 `\item[]`，排版結果將無任何標籤。我們還可以加上字體控制指令。例如，若標籤要變成粗體字的 **A**，指令為 `\item[\textbf{A}]`。標籤也可以是中文字，例如，`\item[{\ctxfr 甲}]`。

如果要改變全部標籤符號，可以使用 `\labelitemi` 或 `\labelitemii` 指令。前者代表最外層之 `item` 指令環境；後者代表次一層之指令環境。再下一層之指令則為 `\labelitemiii` 等等。譬如，文稿中若使用下列指令：

```
\renewcommand{\labelitemii}{+}
```

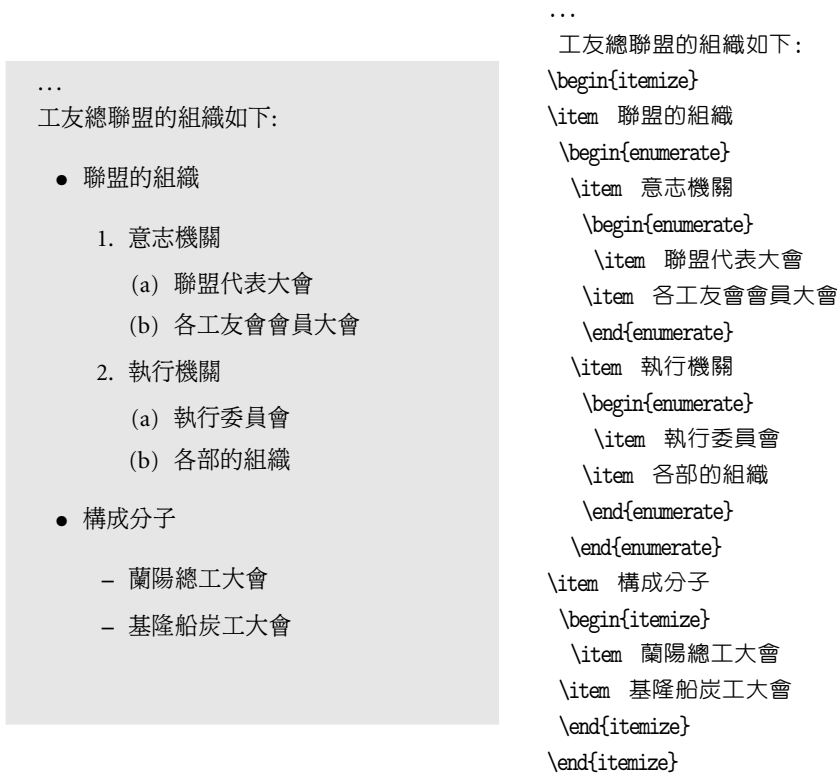


圖 8.1: 條列指令環境

則第2層指令環境之標籤將由 - 改變成 +。

同理, `enumerate` 指令環境也可以用類似的指令變更。此一指令環境之標籤是以數字英文字母依序標出, 控制指令有: `\arabic`, `\roman`, `\Roman`, 與 `\Alph`。譬如,

```
\renewcommand{\labelenumi}{\arabic{enumi}}
```

將把 `enumerate` 指令環境最外層之標籤改變成 1), 2), ... 之形式。同理, 下列一行指令:

```
\renewcommand{\labelenumi}{\Roman{enumi}}
```

把標籤變成大寫羅馬字: I, II, ...。

以上之定義指令若加於全文設定區, 其效果將及於文稿中每一個條列指令環境。反之若指令是下於某指令環境內, 則僅該處之設定改變, 其他指令環境不受

影響。一般而言,改變行距、字體、或其他設定之指令若是下於指令環境中,其影響效果只在該指令環境內。若希望其效果延伸至文稿每一部分,指令應置於全文設定區。

第三種條列指令環境為 `description`。使用此一指令時,必須自行輸入條文標籤符號,並以方括號括起來,請參見底下的例子:

執行委員會權限如下:

- 甲、執行各決議案
- 乙、管理聯盟的財政
- 丙、決定偶發事件之對策

執行委員會權限如下:

```
\begin{description}
\item [甲.] 執行各決議案
\item [乙.] 管理聯盟的財政
\item [丙.] 決定偶發事件之對策
\end{description}
```

標籤符號並不限定是一個字或符號,也可以是一個名詞甚或一段文字:

- 政治 政客玩的遊戲,
- 政客 玩政治的人,
- 政治家 長得好看,而且聲音好聽的政客。

```
\begin{description}
\itemsep=-1pt
\item[{\ctxfb 政治}] 政客玩的遊戲,
\item[{\ctxfb 政客}] 玩政治的人,
\item[{\ctxfb 政治家}] 長得好看,
    而且聲音好聽的政客。
\end{description}
```

其中, `\itemsep=-1pt` 指令的目的是把各列舉項之間的距離縮小一些。

在 `description` 指令環境下,若某條列項之文字超過一行寬度,第2行開頭文字會略往內縮一特定距離,如本例之第3條列項。但是在 `itemize` 與 `enumerate` 指令環境下,列舉條文之長度若超過行寬時,第2行開始每一行之起頭將內縮於標籤相對位置之後,與條文第一行起頭對齊。

條列指令環境可以設定每一層列舉項條文之內縮距離。第一層列舉項之內縮距離是以 `\leftmargini` 控制,最後的字母 `i` 代表第一層。因此,第二層列舉項以 `\leftmarginii` 控制。在圖 8.1 的例子內,我們設定每一層列舉項內縮之距離都是 6mm。亦即,在第一個 `\begin{itemize}` 指令之前加上下列三行指令。

```
\leftmargini=6mm
\leftmarginii=6mm
\leftmarginiii=6mm
```


除了以上三種指令環境外， \TeX 還提供 `list` 指令環境，讓使用者自行設定特別的列舉格式。詳細說明，請見 [Kopka and Daly \(1995\)](#)，頁73–79。

8.5 迷你版面指令環境

文稿版面上，字母或單字是最小的組成要素。一個中文字或英文字母在版面上都是一個小方塊。在橫排文稿中，由單字或字母所串接而成的一行，也是一個方塊。段落是由數行文字組成，在版面上它是一個更大的方塊。因此，文稿版面可以看成是由一堆大大小小的方塊組合而成。

在簡單的文稿中， \TeX 自動把一個一個文字按順序組合成方塊。但有時候我們必須自己控制方塊的大小及位置。舉一個例子來說，如果我們在某一頁的左上角引用一個外製圖形（見第 11 章），其寬度為正常行寬的三分之二，則右邊剩下來的三分之一空白可以用來排版圖形說明文字。因此，圖形是一個大方塊，說明文字是一個小方塊。另外一個例子，在信函結尾處會寫出發信人姓名，其下為其頭銜。假設姓名與頭銜共占 3 行，且排於版面中間靠右之處，我們可以將此 3 行所占空間視為一方塊，再以指令控制其位置。

底下先介紹迷你版面指令，下一小節則介紹文字方塊 (`box`) 指令。

8.5.1 迷你版面

欲將某段文字控制於特定之小方塊中，可以使用 `\parbox` 指令，或者 `minipage` 指令環境。這兩道指令在段落編排上的用途甚廣，值得細心了解。通常，`\parbox` 指令是用於處理較短的段落，`minipage` 指令環境則用於編排較長的文字。最簡單的指令形式如下所示：

```
\parbox[t]{5cm}{...}
```

若使用 `minipage` 指令環境，指令如下：

```
\begin{minipage}[t]{5cm}
...
\end{minipage}
```

其中，`5cm` 設定迷你版面之寬度為 5 公分。

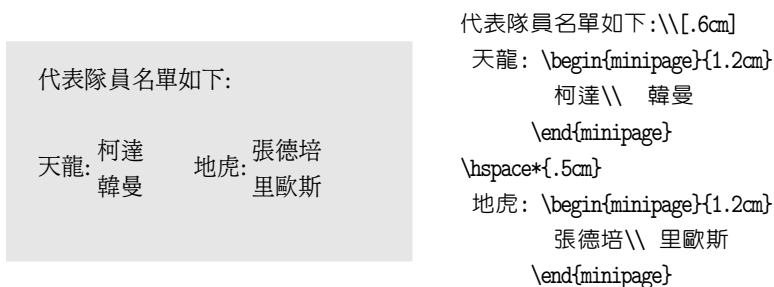


圖 8.2: 迷你版面指令環境例子1

排版時，迷你版面所形成之方塊不會被拆開。換言之，此方塊和一個單字一樣，變成是版面構成的最基本元素。排版時，單字或字母是將其下沿對齊基線 (baseline)。但是，迷你版面之高度可能是單字的數倍，其對齊基線之點可以自行設定。上例中，t 為迷你版面之對齊選項，對齊選項共有三個選擇：

- t 迷你版面之上沿對準基線，
- c 迷你版面之中央對準基線，
- b 迷你版面之下沿對準基線。

其中，c 為內定之選項，可以省略不加。圖 8.2 是使用迷你版面指令環境排版的一個應用例子。

上面說明，迷你版面可以視為一小方塊。因此，本例之隊員名單事實上是由 4 個方塊組成：隊名為其中兩個，其餘兩個方塊各由兩個人名所組成。輸入文字的第一行之後加上 \\[.6cm]，其作用是強迫換行，並加大行距。下一行的基線對齊「天龍」、「地虎」等文字之下緣。兩個迷你版面都沒有選擇對齊點，因此 \TeX 選用內定值，以迷你版面之中央點對齊文字之基線。

若輸入之文字大於迷你版面一行之寬度， \TeX 會先把本行填滿，再把其餘文字排至下一行。在此例子中，我們希望兩個人名上下排出，因此第 1 個人名之後加上換行指令，使第 2 個名字排於下一行。本例中，兩個迷你版面之寬度都是 1.2 公分。實際上，「柯達」兩個字所占寬度大約只有 0.7 公分左右，其後約 0.5 公分在版面上變成空白。萬一我們設定之迷你版面寬度太小容不下三個字，超過長度之文字將移至下一行。

迷你版面內亦可使用段落指令，以 \par 指令或者空白一行為之。但是，新段

代表隊員名單如下:		代表隊員名單如下:\[.5cm]
		\hspace*{.2cm}
隊長	隊員	\begin{minipage}[t]{1.2cm}
陳一軍	山普拉斯	{\ctxfbb 隊長}\ 陳一軍
	阿格西	\end{minipage}
		\hspace{.5cm}
		\begin{minipage}[t]{1.8cm}
		{\ctxfbb 隊員}\ 山普拉斯\ 阿格西
		\end{minipage}

圖 8.3: 迷你版面指令環境例子2

落的頭一行不會內縮 (indent), 這是和全頁版面不同之處。如果希望迷你版面內每一段落開頭也內縮, 可以在文字之前加入以下指令:

```
\parindent=.6cm
```

每一段落開頭內縮之距離將為 0.6 公分。

圖 8.3 是另一個例子, 其中對齊指令之選項為 t。第 2 行的 `\hspace*` 指令使隊員名單右移 0.2 公分。

再舉另一個迷你版面的應用例子。你目前所閱讀的這段文字是納入在一個 `minipage` 指令環境中, 其寬度設定為正常寬度的 0.85, 並靠右編排。請注意, 在迷你版面中段落開頭文字並不內縮。我們所使用之指令如下:

```
\begin{flushright}
\begin{minipage}{.85\textwidth}
...
\end{minipage}
\end{flushright}
```

舊版 $\text{E}_\text{T}_\text{X}$ 中, 迷你版面只能設定對齊基線之位置, 新版則提供另外兩個選項。第一個選項用以設定迷你版面的高度; 第二個選項則設定版面內文字段落之高低位置。因此, 完整的迷你版面指令為:

```
\parbox[ pos ][ height ][ in-pos ]{ width }{ ... }
```

若使用 `minipage` 指令環境, 格式為:

```
\begin{minipage}[ pos][height][in-pos]{width}
...
\end{minipage}
```

第 1 個 *pos* 選項設定迷你版面與基線對齊之位置; 第 2 個 *height* 選項設定版面高度。輸入指令時, 可以直接選擇高度, 如 3cm, 也可以使用下列之單位:

```
\width      文字方塊的寬度,
\height     文字方塊頂端到基線之距離,
\depth     文字方塊的深度,
\totalheight \height 加上 \depth 。
```

譬如說, 若 *height* 設定為 $[0.8\width]$, 則文字方塊之高度為寬度的 0.8。第 3 個 *in-pos* 選項設定文字在迷你版面內之編排位置。選擇 *t* 將使文字排於上端; *b* 排於下端; *c* 則居中編排; *s* 選項則使文字均勻排於迷你版面內。

在迷你版面內, 若下指令改變行距或選用特定英文字體、或作其他設定, 其作用只在指令環境內有效。因此, 若在迷你版面內一開始即選用 *sans serif* 字體。或者特定中文字體, 離開指令環境之後將自動恢復原先之字體。

8.5.2 文字方塊

如上所述, 文稿版面是一個一個方塊組合而成, 上一小節介紹的迷你版面是文字方塊的一種。L^AT_EX 有三種文字方塊的概念: LR 方塊, 段落方塊, 及線條方塊 (*rule boxes*)。在 LR 方塊中文字只能由左至右編排; 段落方塊則是一行一行之文字由上而下垂直疊起來的方塊; 而線條方塊則指由線段本身所構成的方塊。上一小節介紹的迷你版面即為段落方塊, 因為其版面是由一行一行的文字垂直堆積而成。

LR 方塊中之文字只能由左至右水平相接, 指令為:

```
\mbox{...}   \makebox[ width][pos]{...}
\fbbox{...}  \framebox[ width][pos]{...}
```

左邊兩項指令, `\mbox` 指令將大括號內之文字形成一無外框之方塊; `\fbbox` 則自動加上方形外框。例如, `\fbbox{ 文字方塊 }` 指令, 將產生 文字方塊。 `\makebox` 指令類似 `\mbox`, 但可以選擇寬度及方塊內文字之排版方式; 同樣的, `\framebox`

指令類似 `\fbox`，但也可以選擇寬度及方塊內文字之排版位置。例如：

文字方塊	<code>\framebox[4cm][l]{ 文字方塊}</code>
文字方塊	<code>\framebox[4cm][r]{ 文字方塊}</code>

選項 `l` 設定文字靠左，`r` 設定文字靠右。我們還可以選擇 `s` 選項，代表均勻延伸 `stretch`，再使用 `\hfill` 或 `\dotfill` 指令讓文字均勻排於方塊內，例如：

文字	方塊	<code>\framebox[4cm][s]{ 文字\hfill 方塊}</code>
----	----	----------------------------------------------

LR 方塊之 `width` 選項除了直接選定長度之外 (如本例之 `4cm`)，還可以使用下列之單位：`\width`，`\height`，`\depth`，及 `\totalheight`，請參見上一小節之說明。

因為 `\makebox` 或 `\framebox` 為 LR 方塊指令，若文字之長度超過 `width`，排版後文字將凸出方塊外框，不會自動拆為兩行或三行。如果希望自動拆行，可以使用下列指令：

```
\fbox{\begin{minipage}{4cm}
...
\end{minipage}}
```

在 `minipage` 指令環境中之所有文字段落，不管有多少行， \LaTeX 都視為單一字母，因此其外加上 `\fbox` 指令時，即可產生方形外框。

使用文字方塊指令畫方形外框時，可以用 `\fboxrule` 設定外框線條的粗細；而外框與其內文字之距離則以 `\fboxsep` 控制。例如：

文字方塊	<pre>\fboxrule=1pt \fboxsep=15pt \fbox{ 文字方塊}</pre>
------	-----------------------------------------------------

`\fboxrule` 之值若為零，則外框線條將隱而不見。

要將一小段文字上下移動，可以使用下列指令：

```
\raisebox{ lift}[height][depth]{ ... }
```


此道指令將大括號內之文字形成一 `\mbox`，並使之垂直移動 `lift` 之距離。若 `lift` 為

負值, 方塊向下移動。選項 *height* 與 *depth* 分別代表文字方塊之高度與深度。

8.5.3 線條方塊


線條方塊就是一條直線本身, 指令如下:

$$\backslash\text{rule}[lift]{width}{height}$$

其中, *width* 為線條長度, *height* 為其粗細; 而 *lift* 選項則將線條上下移動。因此, $\backslash\text{rule}{1\text{cm}}{0.1\text{cm}}$ 產生: ; 長度為 1 公分, 粗細為 0.1 公分。

在某些應用上, 我們可以將長度或高度設為零。譬如, $\backslash\text{rule}{2\text{cm}}{0\text{pt}}$ 將產生 2 公分的水平空白; 反之, $\backslash\text{rule}{0\text{cm}}{1\text{cm}}$ 將產生 1 公分之垂直空白。底下指令:

$$\backslash\text{framebox}{\backslash\text{rule}{3\text{mm}}{0\text{pt}}\backslash\text{rule}{0\text{pt}}{3\text{mm}}}$$

產生一 3mm 之正方形: 。


8.5.4 儲存方塊

若一組文字在文稿中會重覆出現, 我們可以先將之儲存為方塊, 再於適當地方引用之。儲存方塊之前, 須先取一名字; 新取之方塊名字不可與現有巨集指令名稱相同。方塊命名之指令為 $\backslash\text{newsavebox}$; 儲存方塊可以使用 $\backslash\text{sbox}$ 或者 $\backslash\text{savebox}$ 指令。舉例言之, 若上一小節之正方形取名為 $\backslash\text{sqe}$, 指令為:

$$\backslash\text{newsavebox}{\backslash\text{sqe}}\backslash\text{sbox}{\backslash\text{sqe}}{\backslash\text{framebox}{\backslash\text{rule}{3\text{mm}}{0\text{pt}}\backslash\text{rule}{0\text{pt}}{3\text{mm}}}}$$

利用以上定義, 我們即可使用 $\backslash\text{usebox}$ 指令三次:

$$\backslash\text{usebox}{\backslash\text{sqe}}\backslash\text{usebox}{\backslash\text{sqe}}\backslash\text{usebox}{\backslash\text{sqe}}$$

即可排版出三個連續方塊: 

若使用 $\backslash\text{savebox}$ 指令儲存方塊, 指令為

$$\backslash\text{savebox}{\backslash\text{boxname}}[width][pos]{...}$$

其中, `\boxname` 為自行選定之方塊名稱; `width` 與 `pos` 選項之意義與上一小節所介紹之 `\framebox` 指令選項相同。

8.6 註解與邊註

文稿內排版註解與邊註 (marginal notes) 很容易。註解會自動編上號碼; 其內容以較小之字體排於當頁底下, 邊註則排於版面兩旁。

8.6.1 註解

文稿之註解是以 `\footnote{...}` 指令編排, 此道指令應緊接在引述註解之正文文字或標點符號之後, 中間不留下空白, 而註解內容則輸入於大括號之中。排版之後, 註解內容會出現在該頁正文下方, 正文中將以上標阿拉伯數字標示該註解之編號。若註解內容甚長, 本頁之剩餘空間無法容納, \TeX 會自動將後半部分移至下一頁。

註解內之英文與數字將以較小字體編排, 行距也對應縮小。排版中文稿時, 我們通常須加大行距, 方法之一是在正文開始地方加上 `\fontsize` 指令, 但這只對正文有效, 註解之行距並不受影響。或者, 如果要把全文之行距加大為原來的 1.15 倍, 我們可在全文設定區加上 `\linespread{1.15}` 指令。因為註解之字體較小, 註解內之行距加大為 1.15 倍似乎太大了一些。一個解決的辦法如下: 正文仍以 `\fontsize` 設定行距, 註解之行距則另行控制。在全文設定區鍵入下列指令,

```
\let\oldfootnote\footnote
\renewcommand\footnote[1]{\oldfootnote{%
  \renewcommand\baselinestretch{1.1}%
  \large\footnotesize\ignorespaces#1}}
\addtolength{\footnotesep}{1pt}
```

以上設定將註解之行距拉大為原來的 1.1 倍, 兩個註解之間距則加大 1pt。

8.6.2 註解編號

\TeX 會自動幫註解編上阿拉伯數字號碼。在 `book` 或 `report` 文件類別下, 每一章之註解將重新由 1 起編。但必要時使用者可以自行編號。若要將某一註解自行編號為 4, 指令為 `\footnote[4]{...}`。註解之編號是透過計數器 (counter)。因

此, 另外一個自行編號的方法是透過註解編號計數器 `footnote`。要改變計數器之內容, 可以使用下列指令:

```
\setcounter{footnote}{4}
```

經過以上設定, 下一次再遇到 `\footnote` 指令時, 註解編號將變成 5。

正常情況下, 註解編號為阿拉伯數字。但亦可改為 `*`, `**` 等符號。註解之計數器為 `footnote`, 但實際排版號碼之指令為 `\thefootnote`。L^AT_EX 有一特別用來設定註解符號字體之指令, 稱為 `\fnsymbol`。如果我們使用下列指令:

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}
```

則原來阿拉伯數字 1-9 之註解符號將變成:

```
* † ‡ § ¶ || ** †† ‡‡
```

作此改變之後, 在正文第 9 個註解之後, 使用者應重新將計數歸零, 使下一個註解再從第一個符號起編。如果註解編號是要以 a, b, c, ... 等英文字母排版, 指令為:

```
\renewcommand{\thefootnote}{\alph{footnote}}
```

除了 `\alph` 之外, 尚可使用 `\Roman`, `\roman`, `\Alph` 等指令。

註解將編排於版面下方, 以一橫線與正文隔開。若欲改變此分隔橫線之格式, 可以在全文設定區重新定義:

```
\renewcommand{\footnoterule}{\rule{w}{h}\vspace{-h}}
```

其中, `w` 設定分隔線長度, `h` 為粗細, `\vspace` 指令的作用是調整垂直空間。舉例言之, 若欲取消分隔線, 直接以空白分隔正文與註解, `h` 值應設為 `0pt`。

同一頁中若有兩個以上之註解, 其間隔是以 `\footnotesep` 控制。例如:

```
\footnotesep=5mm
```

指令設定兩個註解之間距為 5mm。

在數學模式或表格中, 不能使用 `\footnote` 指令。萬一有必要在數學式或表格內加上註解, 我們可以使用下列兩道指令間接排版註解:


```
\footnotemark[ n]
\footnotetext[ n]{...}
```

第1道指令可以在正文中加上註解編號， n 為自行選擇之編號；第2道指令則是用於將註解內容排版於版面下方。

舉例來說，若某數學式中須加上註解，因為不能使用 `\footnote` 指令，我們只好以 `\footnotemark` 替代。若不加選項 $[n]$ ，此一註解仍將依續原有之編號。若加上 $[9]$ ，註解編號將變成9。請注意，以上指令純粹用於加註編號，無法編排註解內容。排版完數學式之後，底下再以 `\footnotetext[n]{...}` 指令排版註解內容。註解文字將和其他註解一樣，出現於版面下方。註解文字若有中文，須自行以中文字體指令調整大小。

\LaTeX 會把註解文字排版於版面下方。但是，有些學術期刊則要求把註解內容排版於文稿最後面，此時我們可以使用 John Lavagnino 之 `endnotes` 巨集套件，使用方法請見該套件內附之說明檔。

8.6.3 邊註

註解之內容是排版於版面下方，相對而言，邊註之內容則排版於版面兩側。排版邊註之指令為 `\marginpar`，註解內容將置於一1.9公分寬的迷你版面內，排版於版面之兩旁。因為迷你版面寬度甚小，版面右緣不易對齊。解決此問題的辦法是設定讓每一行依文字之正常寬度排列，右緣不必對齊。

舉一個例子來說，排版本頁邊註之指令如下：

```
... 迷你版面內加上 \verb+\rightskip+ 指令, \marginpar{\sz10
\setlength{\rightskip}{0pt plus 3pt}
邊註之內容將排版於正文方塊旁。}
其作用是讓 ...
```

本例中，我們在迷你版面內加上 `\rightskip` 指令，其作用是讓右沿之空白有一伸縮彈性。本例中，彈性大小為向外延伸3pt。

邊註之內容將排版於正文方塊旁。

邊註之寬度有內定值，但可以變更。本例使用下列指令將邊註文字之寬度設定為1.8公分：

```
\marginparwidth=1.8cm
```

此項更動設定之指令必須下於邊註指令之前。除了寬度之外，我們還可以改變邊註版面與正文方塊之距離，兩個連續邊註之上下距離也可以自行設定。欲將前者距離設為0.5cm，後者設為1cm，指令分別為：

```
\marginparsep=0.5cm
\marginparpush=1cm
```

8.7 照列原文

本書列舉許多例子說明 \TeX 的排版指令。要在文稿內照列排版指令事實上並不容易，譬如要排版數學符號 β ，指令為 β ，但是在文稿內鍵入指令時， \TeX 將直接排版出數學符號。同理，排版 \TeX 標誌符號之指令為 \LaTeX ，但鍵入指令時，標誌符號將直接排版出來，無法看到原始指令。

要將原始排版指令直接列出，可使用 `verbatim` 指令環境，或者 `\verb` 巨集套件。英文 `verbatim` 的意義是照列原文的意思。指令環境 `verbatim` 主要用於排版電腦程式或原始排版指令。在 \TeX 舊2.0版中，`verbatim` 指令環境僅能用於英文，中文無法處理。但從本版開始，中文也能正確處理。除了 `verbatim` 指令環境之外，我們也可使用 Timothy van Zandt 之 `fancyvrb` 巨集套件內之 `Verbatim` 指令環境。以下首先說明 \TeX 照列原文之指令，之後再說明 `fancyvrb` 巨集套件之功能。

在 `verbatim` 指令環境內之任何文字或指令都將照原輸入樣式列出，並選用 `typewriter` 字體排版。譬如，下例中含有 \LaTeX 指令文字，正常情況下排版結果為 \TeX 。但因為全部文字段落置於 `verbatim` 指令環境內，因此指令將照原樣列出，不會排版為 \TeX 標誌符號。

測試 \LaTeX 指令環境，
中英文皆可處理。

```
\ctxfdef{verbatim}{\ctxfr}
\begin{verbatim}
測試 \LaTeX{} 指令環境，
中英文皆可處理。
\end{verbatim}
```

請注意原輸入指令有兩行，排版結果也列為兩行。若使用 `verbatim*` 指令環境，則空白鍵將以 `␣` 符號出現。

若段落內含有中文，我們可以在全文設定區以 `\ctxfdef` 字體指令選用中文字體。本例子是選用圓體編排：

```
\ctxfdef{verbatim}{\ctxfr}
```

如果照列之文字不超過一行,我們可以使用 `\verb` 指令,例如:

```
{\ctxfbb 實質利率}%  
等於名目利率減物價膨脹率。
```

```
\verb+{\ctxfbb 實質利率}%+  
等於名目利率減物價膨脹率。
```

`\verb` 指令之後兩個 `+` 號中間所夾之文字或指令將原文照列出來。如果照列之文字中有 `+` 號,指令分隔符號須改用其他字元,如 `-` 號或 `#` 號。

使用 `verbatim` 指令環境有一些限制,例如不能使用於註解指令之內。要在註解內照列原文,我們可使用 `fancyvrb` 巨集套件。此一巨集套件功能甚強,它除了解決以上所述之問題外,還有其他的功能。譬如,我們可以設定在每一行文字之前編上號碼。`fancyvrb` 巨集套件提供 `Verbatim` 指令環境,此指令環境之基本功能與 \TeX 原有之 `verbatim` 指令環境相同,但是若加上選項,即可產生特別的功能。例如,只加上適當的 `commandchars` 選項,指令環境內每一行可編上編號,也可以處理中文。

圖 8.4 是 `Verbatim` 指令環境的一個例子。首先,選項指令須置於方括號內,本例之選項指令為:

```
commandchars=+\[\]
```

等號右邊定義 3 個符號,第一個符號 `+` 替代 \TeX 之反斜線 `\`; 第二個指令符號 `[` 為左中括號,替代原有之大括號 `{`; 第三個指令符號 `]` 為右中括號,替代 \TeX 原來之右大括號 `}`。

定義以上三個符號之後,原先之反斜線與左右大括號已失去原始之功能,故 \TeX 原有之排版指令,例如 `\textit{test}`, 須更改為 `+textit[test]`。因為反

```
\usepackage{fancyvrb}  
\ctxfr  
欲在 \LaTeX{} 內照列中文,  
可使用 Verbatim 指令環境。  
\begin{Verbatim}[commandchars=+\[\]]  
欲在 \LaTeX{} 內照列中文,  
可使用 Verbatim 指令環境。  
\end{Verbatim}
```

圖 8.4: 照列原文

```

#include <iostream.h>
void main()
{
    char s1[12]="Hello Class";
    char *s2="Hello Class";
    cout << "s1=" << s1 << endl;
    cout << "s2=" << s2 << endl;
    // s1++; 錯誤, s1 為常數
    s2++;
    cout << "s2=" << s2 << endl;
}

\begin{Verbatim}[commandchars=?!?]
#include <iostream.h>
void main()
{
    char s1[12]="Hello Class";
    char *s2="Hello Class";
    cout << "s1=" << s1 << endl;
    cout << "s2=" << s2 << endl;
    // s1++; 錯誤, s1 為常數
    s2++;
    cout << "s2=" << s2 << endl;
}
\end{Verbatim}

```

圖 8.5: fancyvrb 巨集套件

斜線已失去原來定義指令之功能, 故排版時 `\LaTeX{}` 不會產生 \LaTeX 標誌符號, 而是依原鍵入字元列出。

以上例子重新以加號與左右中括號替代原有之指令符號。若 `Verbatim` 指令環境之文字內容含有以上 3 個符號, 替代符號須選用其他字元, 例如 `*!\#`。換言之, 我們以星號替代反斜線, 驚嘆號替代左大括號, `\#` 字號替代右大括號。除了 `commandchars` 指令之外, `Verbatim` 還提供許多控制選項, 例如, `frame=single` 設定在照列出之文字段落加上單一線條之方塊, 方塊與其內文字之間距可以用 \LaTeX 之 `\fboxsep` 指令設定。此外, `numbers=left` 則在每一行文字之前自動編上號碼, 如底下例子所示。

<p>1 欲在 <code>\LaTeX{}</code> 內照列中文,</p> <p>2 可使用 <code>Verbatim</code> 指令環境。</p>	<pre> \usepackage{fancyvrb} \begin{Verbatim}[frame=single,% numbers=left,commandchars=*!\#] 欲在 \LaTeX{ } 內照列中文, 可使用 Verbatim 指令環境。 \end{Verbatim} </pre>
-----------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------

圖 8.5 的例子使用 `Verbatim` 指令環境排版 C 程式語言。因為程式內使用了 `[,]`, 與 `#` 三個符號, 故選用替代符號時須避開。 `fancyvrb` 巨集套件另外一個功能簡化 `\verb` 指令之輸入方式。例如, 以下指令定義兩條垂直線為 `\verb` 指令之範圍:

`\DefineShortVerb{\}`

經過上述定義之後，欲照列 \LaTeX 標誌指令，僅須在指令前後加上垂直線即可：`\LaTeX`，不須使用較複雜的 `\verb+\LaTeX+` 指令。如果照列之文字段落內含垂直線，我們可以使用其它字母如 `#` 替代。此時，上一行指令須更改如下：

`\DefineShortVerb{\#}`

最後，如果 `\verb` 指令要用於註解之內，須在文稿內執行下列宣告指令：

`\VerbatimFootnotes`

另一個具有照列原文功能之巨集套件為 Donald Arseneau 所寫之 `url` 巨集套件。它提供 `\url` 指令。此巨集指令原先設計之目的是用於排版網路網址，但也可以用來照列原文；使用方法與 `\verb` 指令類似。`\url` 指令的使用彈性比 `\verb` 來得大，譬如，前者可置於其他巨集指令之內，後者則不行。`\url` 指令內之文字若遇到一行末端，可以自動拆為兩行，`\verb` 則無換行功能。另外，我們還可以設定是否將 `\url` 指令內所有的空白去除。

通常，`\url` 指令之內容須置於兩個相同的符號之間，但我們也可以將之置放於左右大括號內。請特別注意，`\url` 指令內不能輸入中文。詳細的指令使用說明直接置於 `url.sty` 檔案內，請自行參閱。

8.8 交叉引述

長篇文稿或書籍常在文稿某處引述其他章節或圖表，這稱為 `cross-reference` (交叉引述)。 \LaTeX 會自動對章節編上號碼，欲引述其他章節，首先在被引述章節的標題指令之後下 `\label` 指令作標誌，接下來即可使用 `\ref` 或 `\pageref` 指令引述。譬如，本節標題之排版指令為：

`\section[交叉引述]{交叉引述}\label{refer}`

則文稿任何地方鍵入下列指令：

... 請參見 `\Z\ref{refer}` `\Z` 節 (頁 `\Z\pageref{refer}`) 之說明。

排版之後， \LaTeX 會自動算出章節編號與頁碼：

... 請參見 8.8 節 (頁 149) 之說明。

請注意, 排版時須連續執行 `latex` 兩次才能得到正確結果。除了章節之外, 若將 `\label` 指令置於某註解 `\footnote` 的大括號之內, 文稿內其他地方以 `\ref` 指令即可引述該項註解, `\pageref` 指令則引述出現註解之頁碼。交叉引述指令也可用於數學式中, 請見第 9 章之說明。

8.9 comment 巨集指令

文稿寫作經常須反覆修改增刪。修改過程中, 我們可以使用註銷指令 `%` 暫時去掉某一段落。若段落甚長, 較方便的方法是使用 `comment` 巨集套件, 作者為 Victor Eijkhout。

首先在全文設定區引入巨集套件:

```
\usepackage{comment}
```

接下來, 文稿任何地方可加入 `comment` 指令環境:

```
\begin{comment} ... \end{comment}
```

指令環境內所有之文字與指令, 排版時即略過不處理。我們也可以直接使用下列指令:

```
\comment ... \endcomment
```

其效果相同。進一步的指令細節, 請見 `comment` 巨集套件之說明檔。

8.10 多欄位版面

一般的雜誌常以多欄方式排版。在 `LaTeX` 文件類別指令內加入 `twocolumn` 選項, 即自動以兩欄格式排版。但此法在排版上有一些限制, 譬如, 若文稿前端原以單欄方式編排, 中間改為二欄時, 必須另起一頁。

有鑑於以上之限制, Frank Mittelbach 寫了 `multicol` 巨集套件以供排版多欄位文稿。此巨集套件功能較強, 使用上也很方便。請注意, `multicol` 巨集套件與 `\multicolumn` 指令之名稱接近, 但功能完全不同; 後者是用於排版表格內橫跨多欄之文字。巨集套件 `multicol` 提供 `multicols` 指令環境, 格式如下:

國民所得兩萬美元

台灣的生活水準在提升嗎? 若從大街小巷充斥著賣名牌的商店來看, 答案是肯定的。但如果從生活環境品質日益惡化的現象來看, 答案則是否定的。

三十年前, 台灣的所得尚低。一般民衆最關心如何提升所得; 因此

政府施政也以提升所得為主要目標, 這不難理解。但是, 所得逐漸上升之後, 民衆的偏好、需求會逐漸改變。每一個國家都是如此, 台灣也不例外。最簡單的一個例子, 幾年來出國旅遊風氣日盛, 這表示民衆對休閒的需求日增。

圖 8.6: 多欄式版面

```
\begin{multicols}{ columns}[preface][skip]
...
\end{multicols}
```

其中, *columns* 設定版面之欄位數, *preface* 選項可用以排版多欄位上方之標題。若 *multicols* 指令出現於接近版面下端位置, 則多欄位之文字會跳到下一頁才開始排版。第 3 個選項 *skip* 用於調整空間, 若設為 6 公分, 則當版面所剩空間少於 6 公分時, 多欄位之段落將移至下一頁才開始編排。

底下將第 2 章之例 1 排版為兩欄格式:

```
\usepackage{multicol}
\raggedcolumns
\begin{multicols}{2}[\section*{\ctxfk 國民所得兩萬美元}]
台灣的生活水準在提升嗎?
...
這不難理解。
\end{multicols}
```

排版結果如圖 8.6 所示。

多欄位指令環境可置於其他指令環境內, 例如 *minipage*。事實上, 本例子即置於 *minipage* 指令環境內, 寬度設為原版面寬度的 90%。開始多欄位版面之前, 我們尚加入以下指令:

```
\raggedcolumns
```

此項指令的目的是設定各欄最底下一行不須強迫對齊。如果不下設定指令, 巨集套件將自行加上 `\flushcolumns` 指令, 儘可能讓各欄底端對齊。

多欄位版面之格式可以自行控制。首先, 欄位之間距是由 `\columnsep` 決定, 內定值為 10pt, 使用者可以自行調整。其次, \TeX 開始排版多欄位文字段落之前, 會與前面段落隔出一段距離; 與下接文字之間也會空出一點距離, 此間距是由 `\multicolsep` 指令所控制, 內定值為 12pt, 並有一些伸縮彈性, 必要時可更改之。另外, 欄位之間可以加上垂直線以更明顯區隔。欲加入垂直線, 應在多欄位指令環境之前加入下列指令:

```
\setlength{\columnseprule}{0.4pt}
```

其中, 0.4pt 設定垂直線之粗細。

另外, `\columnbreak` 指令可強迫結束本欄之排版, 底下文字將排版於下一欄。相對而言, 如果我們使用 \TeX 文件類別指令內之 `twocolumn` 選項以排版兩欄式版面, 則結束本欄排版之指令為 `\pagebreak`。除了以上之外, `multicol` 巨集套件尚提供不少指令以控制欄位之編排, 請參見該套件之說明檔。最後, 如果要將多欄位文稿之註解全部排版於右欄下方, 可使用同一作者所寫的 `ftnright` 巨集套件。

9 數學式子

$\text{T}_\text{E}\text{X}$ 以排版數學文稿著稱。Knuth 當初就是爲了排版自己的專業著作，發現一般的排版系統處理數學式子的能力不佳，而開始發展 $\text{T}_\text{E}\text{X}$ 的。 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 是由 $\text{T}_\text{E}\text{X}$ 的巨集指令組合而成，它同樣有排版數學式子的優越能力。在專業排版中，數學式的編排有特別的規範。譬如，上下標須使用較小的字體；數學符號須以數學斜體字編排；符號間距須特別安排；數學函數如 \log , \max 須使用正體字。所有這些細節， $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 都能夠自動處理。

若數式相當複雜，或者其中用到一些符號是 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 所沒有的，我們還可使用美國數學學會所發展的數學符號與巨集套件。本章主要介紹 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 的指令，但也將簡單說明 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}_\text{A}\text{T}_\text{E}\text{X}$ 之功能。欲深入了解數學排版之各種巨集套件，可參考 [Grätzer \(2000\)](#) 或者 [Goossens et al. \(1997\)](#)。

9.1 科技文稿之排版規範

$\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 排版數學或專業文稿的能力甚強，但這並不保證我們的排版一定能達到專業水準。優美排版的條件之一是注意排版規範。排版的一般性規範請見 6.4 節，本節進一步說明排版科技文稿的規範及原則；主要內容係參考 [Goossens et al. \(1997\)](#), 6.1 節。

排版任何文稿首須注意一致性 (consistency)，科學專業文稿更不例外。以數學文稿爲例，大部分的數學符號須使用數學斜體字。排版時，我們只要將符號置於數學模式中， $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 即自動選用數學斜體字。但是，若一個符號在數學式子中是以數學斜體如 MPL 出現，行文中卻以正體字 MPL 或意大利斜體 MPL 出現，讀者可能滿頭霧水。因此，行文中之符號、變數、甚至是數字，必須與出現在數式內者相同。

依排版規範，大部分的數學函數都以正體字編排，例如 \sin , \max 等。這些函數符號可以使用現成的指令排版，如 `\sin`, `\max` 等。相對的，大部分之數學符號

是以斜體排版,但也有例外。例如,度量衡單位通常以正體字排版。為方便參考,底下列出幾項較重要之規範:

- 單位,如 cm, g, KeV 等以正體排版。但,物理學中之常數,如光速 c , 通常以斜體字排版,例如: GeV/c 。
- 化學元素,如 Ne, O, Cu, 通常使用正體字。物理學中基本粒子,如 p, K, g, H 等,亦然。
- 常用之函數,如 \sin , \tan , \min 等,以正體字排版。
- 數字使用正體字。
- 簡寫符號亦使用正體字。譬如,以 exp 代表 experimental; max 代表 maximum; GNP 代表 Gross National Product。
- 微分符號使用正體字,例如 dY/dk 。

9.2 數學環境

數學式可能以兩種型式出現,一是隨文數式 (*in-text formula*); 另一種是數學式自成一或一個段落,我們稱之為展示數式 (*display formula*)。隨文數式是夾在文章中的數學式: 若 $\alpha = 2$, 則 $\alpha^3 = 8$ 。展示數式則是獨立成行的數學式:

$$\phi = \sum_{t=0}^{\infty} \beta^t U(c_t, x_t).$$

排版數式的方法和一般文字不同。欲排版數式,首先須進入數學模式 (*math mode*), 或稱數學環境 (*mathematical environment*)。

隨文數式之數學環境很容易定義,只須在數式前後加上 $\$$ 符號即可,排版時 \TeX 即自動處理所有字體與間距的細節。上列隨文數式的輸入方法如下:

若 $\$ \alpha = 2 \$$, 則 $\$ \alpha^3 = 8 \$$ 。

如果忘了輸入數學模式符號, \TeX 將出現錯誤訊息。欲進入數學模式,除了使用 $\$ \dots \$$ 指令外,另外一種下指令的方法是: $\backslash\text{begin}\{\text{math}\} \dots \backslash\text{end}\{\text{math}\}$; 或者,我們也可以用 $\backslash(\dots \backslash)$ 。顯然,第一種指令型式最簡單,也因此最常使用。

排版展示數式須進入展示數學環境。底下的三種方法都可以達到目的:

```
\begin{equation} ... \end{equation}
\begin{displaymath} ... \end{displaymath}
\[ ... \]
```

若以 `equation` 指令環境排版, 每一行數學式都將自動編上號碼。相對而言, 指令環境 `displaymath` 中之數學式則不編上號碼。第三種指令方式為第二種指令之簡化型式; 我們也可以使用更簡化之指令型式: `$$... $$`。

欲排版上面舉例之展示數式, 指令如下:

```
\[
\phi = \sum^{\infty}_{t=0} \beta^t U(c_t, x_t).
\]
```

輸入以上指令時, 進入與離開數學模式之指令: `\[` 與 `\]` 可以各自單獨占一行, 以方便辨識。展示數式上下, `TeX` 會自動留出適當空白, 不須再空一行。若特別留下空白行, 版面反而不正確。

較複雜的數學式可能橫跨兩行以上, 例如:

$$\begin{aligned}
 m_e &= m_e c \simeq \frac{1}{2} M_e V \\
 &= m_e c^2 \simeq \frac{1}{2} M_e V
 \end{aligned}$$

跨行數式之排版方法, 將在 9.6 節介紹。

9.2.1 數學文稿輸入原則

第 5.3 節曾說明中文稿之輸入原則, 其中特別重要的是在何處換行、中文字間之空白, 及中文與標點符號之間隔。如果排版中文數學文稿, 還須注意以下兩點:

- 隨文數式前後請留一空格。因此, “若 `\alpha` 之值” 是不對的; 正確的輸入法為 “若 `\alpha` 之值”。請注意數學模式前後之空格。
- 展示數式上下不須多留一空行, `TeX` 會自行調整理想間距。

`TeX` 排版數學式子的能力特別強, 但輸入文稿時如果不注意以上的細節, 排版結果會很不理想。

9.2.2 運算符號與上下標

數學式是由數字、數學符號、與加減乘除等運算符號組成。數字與普通運算符號可直接由鍵盤上鍵入。下列符號是直接由鍵盤鍵入：

+ - = < > / : ! ' | [] ()

要注意的是，左右大括號 { } 在 \TeX 中有特殊用途。欲排版左大括號，指令為 \{，右大括號之指令為 \}。利用以上的符號可排版下列數式：

$a < b + c = d $	$\$ a < b+c = d \$$
$y'' = f\{y', y(x)\}$	$\$y'' = f\{y', y(x)\}\$$

其中，二次微分符號須連續鍵入兩個英文引號 ''。若欲排版三次微分符號，則須鍵入三個引號。譬如，鍵入 $\$f'''(x)\$$ ，排版結果為 $f'''(x)$ 。

數學上標是以 ^ 指令輸入，下標是以 _ 指令輸入。因此，輸入 $\$x^{1/2}\$$ 可得到 $x^{1/2}$ ；鍵入 $\$a_{n}\$$ 可得 a_n ；而鍵入 $\$x^2_t\$$ 或者 $\$x_t^2\$$ 均可產生 x_t^2 。最後一個例子顯示，輸入上下標指令之順序並不重要。

上下標若只有單一符號或數字，則大括號可以省略。最後一例中，上下標均為單一變數或數字，輸入時大括號即省略之。輸入上下標時，若忘記加上大括號，很容易發生錯誤。例如，若我們想排版的是 x^{2a} ，但輸入時只鍵入 $\$x^2a\$$ ，排版結果為： x^2a 。底下是幾個較複雜的例子。

x_t^{2a}	$\$x^{2a}_t\$$	$A_{j_{n,m}}^{x_j^2}$	$\$A^{x^2_j}_{j^{2n}_{n,m}}\$$
x_{y^2}	$\$x_{y^2}\$$	$A_{i,j,k}^{-n/2}$	$\$A_{i,j,k}^{-n/2}\$$
x^{y_1}	$\$x^{y_1}\$$		

9.2.3 分式

分式有兩種表現方式。簡單的分式，如 $n/2$ 或者 $m/(m+n)$ ，輸入方法分別為： $\$n/2\$$ 與 $\$m/(m+n)\$$ 。隨文數式之分式通常是以此一方式輸入。較複雜的分式常置於展示數式中。此時，分式的輸入是以 \frac 為之。例如：

$a/(m+n)^2$ 乘上 $1/(x+y)$ 等於

$$\frac{a}{(m+n)^2(x+y)}.$$

$\$a/(m+n)^2\$$ 乘上 $\$1/(x+y)\$$ 等於

```
\[
\frac{a}{(m+n)^2(x+y)}.
\]
```

如本例所示，分式指令 `\frac` 包含兩項數式，第一項為分子，第二項為分母。兩項都須以大括號括起來。本例亦顯示，展示數式指令之前並不須空一行，`ETEX` 會在正文與數式之間加入適當間距。

輸入複雜的分式時，我們應在適當的地方加上大括號，否則容易出現錯誤。底下是展示數式之分式：

$$\frac{x+y}{1+\frac{y}{x+y}}$$

```
\[
\frac{x+y}{1+\frac{y}{x+y}}
\]
```

分式指令 `\frac` 也可以用於行文中，以產生上下層的分式，如 $\frac{x}{x+y}$ 。但如此一來版面會顯得擁擠，因此此種用法比較少見。

9.2.4 開根號

排版開根號的指令很簡單。譬如， $\sqrt{x+y}$ 的指令是 `\sqrt{x+y}`。開根號上面的橫線涵蓋的範圍是大括號內所有的符號。如果誤將左右大括號輸入成圓括號，你等於是輸入 `\sqrt{(x+y)}`，排版結果將變成 $\sqrt{(x+y)}$ ，根號的橫線只涵蓋了左圓括號。

欲排版三次方以上的開根號，`\sqrt` 指令之後必須加入次方選項。例如，排版 $\sqrt[3]{x+y}$ 的指令是：`\sqrt[3]{x+y}`。底下是一個較複雜的例子：

$$u = \sqrt[3]{-q + \sqrt{q^2 + p^3}}$$

```
\[
u = \sqrt[3]{-q+\sqrt{q^2+p^3}}
\]
```

本例中，數學模式指令直接置於數式之前後。為了易於分辨，數學模式指令也可以單獨成一行，數學式則輸入於兩行指令中間。

表 9.1: 希臘字母符號

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	τ	<code>\tau</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	φ	<code>\varphi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	χ	<code>\chi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>	ω	<code>\omega</code>
η	<code>\eta</code>	ξ	<code>\xi</code>				
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

9.3 數學符號

數學文稿中經常使用各種數學符號，這些符號必須以指令輸入，而且必須置於數學環境中。本節將依序介紹各種數學符號之指令。

9.3.1 希臘字母符號

數學文稿經常使用希臘字母符號， \TeX 直接採用希臘字母之音標為其指令名稱，很容易記住。表 9.1 列出所有的希臘字母符號。利用表中所列指令，欲排版 X_b^β 須鍵入： `$\$X^\beta_b\$$` 。欲排版：

$$(\gamma^\mu - m)\psi = 0.$$

須鍵入： `$\$(\gamma^\mu - m)\psi = 0.\$$` 。在數學模式中， \TeX 會自行調整各符號的間距。鍵入文稿時，符號指令之間留不留空白對於排版結果並無影響。因此，輸入上例時若將空白全部去除： `$\$(\gamma^\mu - m)\psi = 0.\$$` ，排版結果相同。

除了希臘字母之外， \TeX 另外有 26 個大寫字母的數學花體字 (script letters 或 calligraphic):

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

排版指令為 `\mathcal`。譬如，欲排版 \mathcal{R}^n ，應鍵入 `$\$\mathcal{R}^n\$$` 。但如果使用舊 2.09 版 \TeX ，指令為 `\cal`。

數學符號也可以改變字體。上面所介紹的 `\mathcal` 指令是選用數學字體的一個例子。在 \TeX 中，選用數學字體之指令包括：

`\mathrm` `\mathtt` `\mathbf`
`\mathsf` `\mathit` `\mathcal`

數學字體指令只改變英文字母、數字、大寫希臘字母；其他符號，如小寫希臘字母或下一小節所介紹的函數符號，並不受影響，請見以下的例子：

$A = B^0(x)$	\[\n\mathbf{A}=\mathbf{B}^0(x)\n\]
--------------	------------------------------------

9.3.2 粗體數學符號

如上所述，`\mathbf` 僅把英文、數字、與希臘字母變為粗體字，對其他的數學符號沒有作用。欲排版粗體數學符號，可使用 `bm` 巨集套件。引入套件之後，以 `\bm` 指令即可排版粗體符號。譬如，

$\alpha/\beta \neq \alpha/\beta$	<code>\bm{\alpha/\beta} \neq \alpha/\beta</code>
----------------------------------	--------------------------------------------------

另一個方法是使用 `amsmath` 巨集套件。引入巨集套件之後，排版粗體數學符號的指令為 `\boldsymbol`。例如，`\boldsymbol{\alpha}` 可排版 α 。

如果要將式子中全部符號、變數等都變為粗體字，可使用 `\boldmath` 指令。欲取消數學粗體字，指令為 `\unboldmath`。請注意，這兩道指令都必須下於數學模式之外，否則將出現錯誤。

$A = B^0(x)$	<code>\boldmath</code>
$A = B^0(x)$	\[\n A=B^0(x) \n\]
$A = B^0(x)$	<code>\unboldmath</code>
$A = B^0(x)$	\[\n A=B^0(x) \n\]

有些數學符號並無粗體字形，若加上數學粗體字指令， \TeX 會設法以普通字體替代之。

表 9.2: 函數符號

<code>\arccos</code>	<code>\arcsin</code>	<code>\arctan</code>	<code>\arg</code>	<code>\cos</code>
<code>\cosh</code>	<code>\cot</code>	<code>\coth</code>	<code>\csc</code>	<code>\deg</code>
<code>\det</code>	<code>\dim</code>	<code>\exp</code>	<code>\gcd</code>	<code>\hom</code>
<code>\inf</code>	<code>\ker</code>	<code>\lg</code>	<code>\lim</code>	<code>\liminf</code>
<code>\limsup</code>	<code>\ln</code>	<code>\log</code>	<code>\max</code>	<code>\min</code>
<code>\Pr</code>	<code>\sec</code>	<code>\sin</code>	<code>\sinh</code>	<code>\sup</code>
<code>\tan</code>	<code>\tanh</code>			

9.3.3 函數符號

依據專業排版規範，數學變數應以數學斜體字編排，但函數則應該以正字體排版。函數符號，如 `log`，`max` 等，若直接輸入，如 `$log xy$`，排版結果為 $\log xy$ ，其中，`log` 函數及數學變數 xy 都以數學斜體字排版，且兩者之間並未留有適當空白。在 \TeX 中，函數應以指令輸入：`$\log xy$`，排版結果為 $\log xy$ 。表 9.2 列出函數符號指令，以供參考。

統計學中經常使用數學期望值 (expectation) 與變異數 (variance) 符號。根據排版規範，函數符號應使用正體字，因此排版結果應為 $E(x)$ 與 $\text{var}(x)$ 。這兩個函數 \TeX 並無現成的指令，因此我們須自行控制。在數學模式中，排版正體字可使用 `\mbox` 指令。因此以上兩個式子之輸入指令分別為：`$\mbox{E}(x)$` 與 `$\mbox{var}(x)$`。如果文稿中經常使用這兩個符號，我們可以定義兩個巨集指令，以節省輸入時間，並避免錯誤。巨集指令之定義方法，請見第 14 章之說明。

9.3.4 積分與加總函數

某些數學符號在隨文數式中會比在展示數式中小一些，積分與加總函數是兩個例子。輸入積分符號之指令為 `\int`，加總符號指令為 `\sum`。在行文當中，積分函數之排版為 $\int_0^1 f(x)dx$ ，加總函數為 $\sum_{i=1}^n x_i$ 。若是獨立一行之展示數式，排版結果為：

$\int_0^1 f(x)dx \text{ and } \sum_{i=1}^n x_i$	<pre>\begin{equation*} \int^1_0 f(x)dx \mbox{ and } \sum^n_{i=1} x_i \end{equation*}</pre>
-------------------------------------------------	--------------------------------------------------------------------------------------------

表 9.3: 積分與加總符號

\sum	<code>\sum</code>	\prod	<code>\prod</code>	\int	<code>\int</code>	\oint	<code>\oint</code>
\coprod	<code>\coprod</code>	\bigcap	<code>\bigcap</code>	\bigcup	<code>\bigcup</code>	\bigvee	<code>\bigvee</code>
\bigwedge	<code>\bigwedge</code>	\odot	<code>\odot</code>	\otimes	<code>\otimes</code>	\oplus	<code>\oplus</code>
\biguplus	<code>\biguplus</code>	\bigsqcup	<code>\bigsqcup</code>				

除了符號大小有所差異之外，函數上下標之位置也不同。在隨文數式中，爲了不占太大的空間，上下標是置於符號本身的右上方及右下方；展示數式則置於上下方。如果要改變設定，希望隨文數式中上下標也置於正上下方，可以使用 `\limits` 指令。例如，以下指令 `\sum\limits^{t=0}` 可排版出： $\sum_{t=0}^t$ 。請注意，使用 `\limits` 指令時，上下行距會自動加大一些，效果不見得好。相反的，若展示數式中之上下標希望置放於右上下方，則將 `\nolimits` 指令加在上下標指令之前即可。

除了積分與加總函數之外，表 9.3 所列各函數都會因隨文數式或展示數式而調整其排版方式。

9.3.5 箭號與相對關係符號

相對關係符號是指諸如大於或等於之類的符號，數學文稿中經常使用。以下，我們分箭號 (arrow symbols)，相對關係符號 (relation symbols) 與二元運算符號 (binary operators) 三大類分別介紹之。

表 9.4 列出箭頭符號。這些符號除了用於數學式子之外，也可以和直線、橫線連接，用於排版線條形、流程圖等等。其中， \leftarrow 符號之指令爲 `\leftarrow`，但也可以使用較簡單的 `\gets` 指令。同樣的， \rightarrow 右箭頭符號之指令爲 `\rightarrow`，但可以使用較簡化的 `\to` 指令。如果是要排版向量符號，也可直接使用 $\overrightarrow{x+y}$ 指令，排版結果爲： $\overrightarrow{a+b}$ 。反之，`\overleftarrow{x+y}` 指令可排出 $\overleftarrow{x+y}$ 。表 9.4 列舉 \TeX 系統所提供的箭號指令。

雖然由表 9.4 之指令已能排版各式各樣的箭號，但是再多的符號似乎也永遠無法滿足所有人的需求。爲了應付數學文稿排版的需求，美國數學學會 (American Mathematical Society) 又設計一些符號，並提供 `amssymb` 巨集套件以方便引用。譬如，一般的文稿中也可能使用 \because 符號代表「因爲」，以 \therefore 符號表示「所

表 9.4: 箭頭符號 (arrow symbols)

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightrightarrows	<code>\rightrightarrows</code>	\leadsto	<code>\leadsto</code>		

表 9.5: AMS 箭號 (amssymb 巨集套件)

\Rightarrow	<code>\Rightarrow</code>	\rightsquigarrow	<code>\rightsquigarrow</code>	\leftleftarrows	<code>\leftleftarrows</code>
\leftrightarrow	<code>\leftrightarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>	\twoheadleftarrow	<code>\twoheadleftarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\rightleftarrows	<code>\rightleftarrows</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>	\looparrowleft	<code>\looparrowleft</code>
\looparrowright	<code>\looparrowright</code>	\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>	\circlearrowleft	<code>\circlearrowleft</code>
\circlearrowright	<code>\circlearrowright</code>	\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>
\uparrows	<code>\uparrows</code>	\downdownarrows	<code>\downdownarrows</code>	\upharpoonleft	<code>\upharpoonleft</code>
\upharpoonright	<code>\upharpoonright</code>	\restriction	<code>\restriction</code>	\downharpoonleft	<code>\downharpoonleft</code>
\downharpoonright	<code>\downharpoonright</code>	\multimap	<code>\multimap</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>
\nleftarrow	<code>\nleftarrow</code>	\nrightarrow	<code>\nrightarrow</code>	\nLeftarrow	<code>\nLeftarrow</code>
\nrightarrow	<code>\nrightarrow</code>	\nleftrightarrow	<code>\nleftrightarrow</code>	\nLeftrightarrow	<code>\nLeftrightarrow</code>

以 \Leftarrow ，但 \TeX 並未提供這兩個符號。`amssymb`巨集套件則提供這兩個符號，故若在全文設定區引入`amssymb`巨集套件：

```
\usepackage{amssymb}
```

文稿中數學模式內即可使用`\because`與`\therefore`指令排版以上這兩個符號。

巨集套件`amssymb`提供的符號可簡單區分為5類，以下將依序介紹。首先，AMS箭頭符號全部列於表9.5以與 \TeX 原有的箭號對照。其次，相對關係符號列於表9.6。相對關係符號中，`\leq`指令用以排版 \leq ，但指令簡化為`\le`；而`\geq`指

表 9.6: 相對關係 (relation) 符號

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>	\mid	<code>\mid</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\Join	<code>\Join</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\neq	<code>\neq</code>	\smile	<code>\smile</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\doteq	<code>\doteq</code>	\frown	<code>\frown</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>	\vdash	<code>\vdash</code>
\dashv	<code>\dashv</code>	\notin	<code>\notin</code>				

令可以簡化為 `\ge`。另外， \neq 之指令為 `\neq`，但是，`\not=` 也產生同樣的結果。最後一個例子說明，在相對關係符號指令之前加上 `\not` 指令，將產生否定之關係符號。譬如，鍵入以下指令 `\not\in` 與 `\not>`；排版結果分別為 \notin 與 $\not>$ 。不過，`\not\in` (\notin) 與 `\notin` (\notin) 排版結果並不完全相同；排版專家認為後者較佳。

表 9.6 所列符號大部分是 T_EX 系統所原有的。但是，其中的 \sqsubset 與 \sqsupset 是 L^AT_EX 特別製造之符號，指令分別為 `\sqsubset` 與 `\sqsupset`。欲使用這兩個符號，我們必須在全文設定區引用 `latexsym` 巨集套件：

```
\usepackage{latexsym}
```

若不事先引入巨集套件，文稿中使用這兩個符號指令時會出現符號指令未曾定義 (undefined) 之錯誤訊息。

表 9.6 中雖然有大於與小於符號，但有些文稿使用大於、小於並列符號。我們可以由以上兩個符號合併而成。在全文設定區定義下列指令：

```
\def\gtls{\mbox{$_{<}$}\llap{\mbox{^>}$}}
```

接下來在文稿內輸入 `$\gtls b$` 即可排版： $a \gtrsim b$ 。不過，直接以 `amssymb` 巨集套件所提供之符號排版，品質更佳。表 9.7 列出 AMS 之雙元關係符號，其中排版大於、小於並列符號之指令為：`\gtrless`。類似的符號還有很多，譬如大於、等於、小於並列符號之指令為：`\gtreqless`。

除了雙元關係符號之外，AMS 另有 negated binary symbols，我們稱之為負雙

表 9.7: AMS 雙元關係符號 (amssymb 巨集套件)

\leqq	<code>\leqq</code>	\leqslant	<code>\leqslant</code>	\leqslantless	<code>\leqslantless</code>
\lesssim	<code>\lesssim</code>	\lessapprox	<code>\lessapprox</code>	\approxq	<code>\approxq</code>
\geqq	<code>\geqq</code>	\geqslant	<code>\geqslant</code>	\geqslantgtr	<code>\geqslantgtr</code>
\gtrsim	<code>\gtrsim</code>	\gtrapprox	<code>\gtrapprox</code>	\eqsim	<code>\eqsim</code>
\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\lll	<code>\lll</code>
\ggg	<code>\ggg</code>	\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\lesseqgtr	<code>\lesseqgtr</code>
\gtreqless	<code>\gtreqless</code>	\doteqdot	<code>\doteqdot</code>	\Doteq	<code>\Doteq</code>
\eqcirc	<code>\eqcirc</code>	\risingdotseq	<code>\risingdotseq</code>	\circeq	<code>\circeq</code>
\fallingdotseq	<code>\fallingdotseq</code>	\triangleq	<code>\triangleq</code>	\backsim	<code>\backsim</code>
\thicksim	<code>\thicksim</code>	\backsimeq	<code>\backsimeq</code>	\thickapprox	<code>\thickapprox</code>
\subseteqq	<code>\subseteqq</code>	\supseteqq	<code>\supseteqq</code>	\Subset	<code>\Subset</code>
\Supset	<code>\Supset</code>	\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\curlyeqprec	<code>\curlyeqprec</code>
\curlyeqsucc	<code>\curlyeqsucc</code>	\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\vartriangleright	<code>\vartriangleright</code>	\shortparallel	<code>\shortparallel</code>	\Vdash	<code>\Vdash</code>
\vartriangleleft	<code>\vartriangleleft</code>	\smallsmile	<code>\smallsmile</code>	\Vvdash	<code>\Vvdash</code>
\trianglerighteq	<code>\trianglerighteq</code>	\shortmid	<code>\shortmid</code>	\smallfrown	<code>\smallfrown</code>
\trianglelefteq	<code>\trianglelefteq</code>	\bumpeq	<code>\bumpeq</code>	\between	<code>\between</code>
\blacktriangleleft	<code>\blacktriangleleft</code>	\pitchfork	<code>\pitchfork</code>	\varpropto	<code>\varpropto</code>
\blacktriangleright	<code>\blacktriangleright</code>	\Bumpeq	<code>\Bumpeq</code>	\backepsilon	<code>\backepsilon</code>
\therefore	<code>\therefore</code>	\because	<code>\because</code>		

表 9.8: AMS 負雙元關係符號 (amssymb 巨集套件)

\nless	\nleq	\nleqslant
\nleqq	\lneq	\lneqq
\ngtr	\ngeq	\ngeqslant
\ngeqq	\gneq	\gneqq
\lvertneqq	\gvertneqq	\lnsim
\gnsim	\lnapprox	\gnapprox
\nprec	\nsucc	\npreceq
\nsucceq	\precneqq	\succneqq
\precnsim	\succnsim	\precnapprox
\succnapprox	\nsim	\ncong
\nmid	\nshortmid	\nshortparallel
\nparallel	\ntrianglelefteq	\nvDash
\nVDash	\nVDash	\ntriangleleft
\ntriangleright	\nvdash	\ntrianglerighteq
\nsubseteq	\nsupseteq	\nsubseteqq
\nsupseteqq	\subsetneq	\supsetneq
\varsubsetneq	\varsupsetneq	\subsetneqq
\supsetneqq	\varsubsetneqq	\varsupsetneqq

元關係符號。這些符號全部列於表 9.8, 以供參考使用。此表內之符號有很多是表 9.7 內符號之相反。譬如, `\leqq` 指令產生 \leqq 符號; 而 `\nleqq` 則產生 \nleqq 。

接下來, 表 9.9 列出雙元運算符號。其中, `\times` 用以排版乘號, `\div` 用以排版除號。即使是一般文稿, 也常有機會使用加減乘除符號。在 \LaTeX 中, 使用加減乘除應先進入數學模式。因此, 欲排版加號 $1 + 2 = 3$, 指令為: `$1+2=3$`。若未進入數學模式, 排版結果為 $1+2=3$ 。減號亦然, `$3-1=2$` 排版結果為 $3 - 1 = 2$ 。若未進入數學模式, 排版結果為: $3-1=2$ 。除了 \TeX 的雙元運算符號外, AMS 又提供更多的雙元運算符號, 如表 9.10 所示。其中, `\centerdot` 比 \LaTeX 之 `\cdot` 要稍大一些。

請注意, 在數學式子內, `\dagger` 指令可排版 \dagger 符號; `\ddagger` 指令可排版 \ddagger

表 9.9: 雙元運算符號

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\oplus	<code>\oplus</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\triangleup	<code>\bigtriangleup</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	∇	<code>\bigtriangledown</code>	\otimes	<code>\otimes</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\vee	<code>\vee</code>	\triangleleft	<code>\lhd</code>	\bigcirc	<code>\bigcirc</code>
\circ	<code>\circ</code>	\wedge	<code>\wedge</code>	\triangleright	<code>\rhd</code>	\dagger	<code>\dagger</code>
\bullet	<code>\bullet</code>	\setminus	<code>\setminus</code>	\triangleleft	<code>\unlhd</code>	\ddagger	<code>\ddagger</code>
\cdot	<code>\cdot</code>	\wr	<code>\wr</code>	\triangleright	<code>\unrhd</code>	\amalg	<code>\amalg</code>

表 9.10: AMS 雙元運算符號 (amssymb 巨集套件)

$\dot{+}$	<code>\dotplus</code>	\ltimes	<code>\ltimes</code>	\smallsetminus	<code>\smallsetminus</code>
\rtimes	<code>\rtimes</code>	\Cup	<code>\CAP, \doublecap</code>	\leftthreetimes	<code>\leftthreetimes</code>
\Cup	<code>\Cup, \doublecup</code>	\rightthreetimes	<code>\rightthreetimes</code>	$\bar{\wedge}$	<code>\barwedge</code>
\curlywedge	<code>\curlywedge</code>	\veebar	<code>\veebar</code>	\curlyvee	<code>\curlyvee</code>
\doublebarwedge	<code>\doublebarwedge</code>	\boxminus	<code>\boxminus</code>	\circleddash	<code>\circleddash</code>
\boxtimes	<code>\boxtimes</code>	\circledast	<code>\circledast</code>	\boxdot	<code>\boxdot</code>
\circledcirc	<code>\circledcirc</code>	\boxplus	<code>\boxplus</code>	\centerdot	<code>\centerdot</code>
\divideontimes	<code>\divideontimes</code>	\intercal	<code>\intercal</code>		

符號。但在非數學模式內，我們可使用 `\dag` 與 `\ddag` 指令排版同樣的符號。

除了以上各表所列之外， $\text{T}_\text{E}\text{X}$ 還有一些不易歸類的符號。有些用於數學或自然科學中，有些是音符，有些則用於一般文稿。我們將它們全部列於表 9.11。此表中，下列 9 個符號是由 $\text{E}\text{T}_\text{E}\text{X}$ 的 `latexsym` 巨集套件所提供：

$\square \diamond \cup \bowtie \triangleleft \triangleright \trianglelefteq \trianglerighteq \sim$

連同前面所介紹的 \sqsubset 與 \sqsupset ，`latexsym` 巨集套件共提供 11 個符號。欲使用這些符號，須先引用 `latexsym` 巨集套件。或者，若引用了 `amssymb` 巨集套件，我們也可以使用這些指令。

表 9.11 列有幾個樂譜符號，如 \sharp 與 \flat 等，這些當然不足以排版樂譜。不過，近幾年來陸續有人發展幾套以 $\text{T}_\text{E}\text{X}$ 系統為基礎的樂譜排版系統；請見 [Goossens et al. \(1997\)](#) 之介紹。AMS 所提供的其他符號列於表 9.12，其中，`\square` 可排版

表 9.11: 其他符號

\aleph	<code>\aleph</code>	$'$	<code>\prime</code>	\forall	<code>\forall</code>	∞	<code>\infty</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\square	<code>\Box</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\diamond	<code>\Diamond</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\triangle	<code>\triangle</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\clubsuit	<code>\clubsuit</code>
\wp	<code>\wp</code>	\perp	<code>\perp</code>	\sharp	<code>\sharp</code>	\diamondsuit	<code>\diamondsuit</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\backslash	<code>\backslash</code>	\heartsuit	<code>\heartsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	∂	<code>\partial</code>	\spadesuit	<code>\spadesuit</code>
\mho	<code>\mho</code>						

表 9.12: AMS 希臘字母與其他符號 (AMS miscellaneous)

\hbar	<code>\hbar</code>	\backprime	<code>\backprime</code>	\hslash	<code>\hslash</code>
\varnothing	<code>\varnothing</code>	\vartriangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>
\triangledown	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\square	<code>\square</code>
\blacksquare	<code>\blacksquare</code>	\lozenge	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>
\textcircled{S}	<code>\textcircled{S}</code>	\bigstar	<code>\bigstar</code>	\angle	<code>\angle</code>
\sphericalangle	<code>\sphericalangle</code>	\measuredangle	<code>\measuredangle</code>	\nexists	<code>\nexists</code>
\complement	<code>\complement</code>	\mho	<code>\mho</code>	\eth	<code>\eth</code>
\Finv	<code>\Finv</code>	\diagup	<code>\diagup</code>	\Game	<code>\Game</code>
\diagdown	<code>\diagdown</code>	\Bbbk	<code>\Bbbk</code>		
\digamma	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>
\daleth	<code>\daleth</code>	\gimel	<code>\gimel</code>	\lrcorner	<code>\lrcorner</code>
\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>

一正方形: \square 。若希望方形稍小一些, 可使用 `latexsym` 巨集套件提供的 `\Box` 指令, 排版結果為 \square 。另外, `\blacksquare` 可排版一實心小正方形: \blacksquare , 有人用於標示數學證明結束。

最後, 數學中之 `mod` 函數有兩種型式, 第一種為雙元型式 (binary), 輸入指令為 `\bmod`; 第二種為括號型式 (parenthesized), 指令為 `\pmod`。譬如, 輸入 `\a\bmod b` 指令結果為: $a \bmod b$; `\pmod{a+b}` 指令則排版為: $(\bmod a + b)$ 。

如果只是偶而排版數學式, 要記得以上這些符號之指令其實並不容易。此時, WinEdt 軟體之數學符號圖形功能很有幫助。點選 WinEdt 視窗上之 Σ 圖

像 (icon), 即可開啓數學符號工作列, 其上列出各種數學符號之圖樣。點選任一圖樣, WinEdt 即在文稿中輸入指令, 甚為方便。使用完畢, 再點選 Σ 圖像, 即關閉工作列。

9.3.6 數學重音符號、底線與上線

以上所介紹的數學符號可附加上重音符號 (accents)。例如, $\hat{\beta}$ 或者 \vec{a} 。在數學指令環境中, 重音附加符號共有下列幾種:

\hat{o}	<code>\hat{o}</code>	\acute{o}	<code>\acute{o}</code>	\bar{o}	<code>\bar{o}</code>	\dot{o}	<code>\dot{o}</code>	
\check{o}	<code>\check{o}</code>	\grave{o}	<code>\grave{o}</code>	\vec{o}	<code>\vec{o}</code>	\ddot{o}	<code>\ddot{o}</code>	重音符
\breve{o}	<code>\breve{o}</code>	\tilde{o}	<code>\tilde{o}</code>					

號指令當然也可以加在希臘字母上面, 如 $\tilde{\beta}$ 。另外, `\hat` 符號若要加大一些, 可使用 `\widehat`。譬如, 欲排版 $\widehat{x + y/x}$, 輸入指令為:

`\widehat{x+y}/\widehat{xy}`

同樣的, 較大的 `\tilde` 指令是 `\widetilde`。

在專業排版中, 字母 i 與 j 之上若有附加符號, 其頂上之小點應去掉。要排版去掉小點的字母 i 與 j , 指令分別為 `\imath` 與 `\jmath`。例如,

`\mathbf{i+j \neq \vec{\imath} + \bar{\jmath}}`

排版結果為 $i + j \neq \vec{i} + \bar{j}$ 。

重音符號是在字母上面加符號, 與此類似的是底線與上線。在普通文字稿中, 我們使用 `\underline` 指令畫底線, 此一指令也可以用於數式中:

value is $\underline{3x}$.	value is <code>\underline{3x}</code> .
-----------------------------	----------------------------------------

若要加入上線, 我們可以使用 `\overline` 指令。例如:

$\overline{\overline{x^2+1}}$	<code>\overline{\overline{x^2+1}}</code>
$\overline{x^2+1}$	<code>\overline{x^2+1}</code>

底線與上線是在符號的上下加一直線。有時候，我們須加上大括號。上括號之指令為 `\overbrace`，底括號為 `\underbrace`。例如：

$\overbrace{a + \underbrace{b + c}}$	<code>\overbrace{a+\underbrace{b+c}}</code>
--------------------------------------	---------------------------------------------

以上是隨文數式例子。展示數式中，上括號與底括號若再加上下標，排版結果如下所示：

$\underbrace{a + \overbrace{b + \cdots + x + y + z}^{24}}_{26}$	<code>\[</code> <code>\underbrace{a + \overbrace{b</code> <code>+ \cdots + x + y}^{24} + z}_{26}</code> <code>\]</code>
-----------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------

9.3.7 上下重疊符號

要把一個符號疊在另一個符號的上面，可以使用 `\stackrel` 指令：

$A \xrightarrow{a} B$ $\vec{x} \stackrel{\text{def}}{=} (x_1, \dots, x_n)$	<code>\$A \stackrel{a}{\rightarrow} B\$</code> <code>\$\$\vec{x} \stackrel{\text{def}}{=} (x_1, \dots, x_n)\$\$</code>
----------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------

`\stackrel` 指令有兩項變數，排版之後第一項變數的字體將會縮小一些，並移至第二項變數的上頭。

我們也可以直接使用 $\text{T}_\text{E}\text{X}$ 所提供的指令排版上下重疊之數學符號。第一個指令是 `\choose`，應用例子是排版 binomial coefficient。例如：

$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$	<code>\[</code> <code>{n+1 \choose k} =</code> <code>{n \choose k} + {n \choose k-1}</code> <code>\]</code>
--------------------------------------------------	----------------------------------------------------------------------------------------------------------------------

請注意，`\choose` 指令會自動加上左右圓括號。

第二個指令是 `\atop`。顧名思義，此一指令的功能是把一個符號加到另一個符號上面。譬如，

$$\sum_{\substack{0 \leq i \leq n \\ 0 \leq t}} U(c_{it}, x_{it})$$

```
\[
\sum_{\scriptstyle 0 \leq i \leq n
\atop \scriptstyle 0 \leq t}
U(c_{it}, x_{it})
\]
```

此例子中之 `\scriptstyle` 指令是用以控制下標符號之大小。若不加這項指令， \TeX 將以內定字體大小編排，結果會顯得太小。

統計學或計量經濟學的論文中經常出現迴歸方程式 (regression equation)，其中估計值的下方須列出標準差，譬如：

$$y = 1.23 x_1 + 2.34 x_2 + \epsilon$$

(2.23) (1.22)

欲排版此式，我們可以使用下列指令：

```
\[
y = \tb{1.23}{(2.23)}x_1 + \tb{2.34}{(1.22)}x_2 + \epsilon
\]
```

其中，`\tb` 為一巨集指令，其定義如下：

```
\def\tb#1#2{\mathop{#1\phantom{\sum}}\limits_{\displaystyle #2}}
```

此項巨集指令須在數學環境內才能使用。欲使用 `\tb` 巨集指令，我們須在全文設定區鍵入其定義。

使用 `\tb` 指令時必須填入兩項參數，第一項為迴歸式之數字，如上式中之 1.23，第二項為排版於其下之數字，如上式中之 (2.23)。第二項數字與數學式之距離可以自行調整。上面巨集指令的定義中使用了數學符號 `\sum`。該符號之高度即為上下兩行之距離。若希望距離加大一些，只要使用較大的符號，如 `\big(` 或者 `\Big(`，替代定義式中之 `\sum` 即可。

9.3.8 連續點

數學文稿中，有時候以三個連續的圓點表示中間省略的符號。排版連續點之指令為 `\ldots` 或者 `\cdots`。前項指令所產生的三點位於基線 (baseline) 上；後者的點位置則稍微高一些。例如，

把 x_1, x_2, \dots 各項相加, 得 $X = x_1 + x_2 + \dots$	把 $\$x_1, x_2, \dots\$$ 各項相加, 得 $\$X=x_1+x_2+\dots\$$
-----------------------------------------------------	-------------------------------------------------------

本例中, `\cdots` 所產生的連續點, 其位置稍高於 `\ldots`。相對的, `\cdots` 通常是用於 $+$, $-$, 與 $=$ 號之間。變數之間, 如 a, b, \dots , 或者逗點之後之連續點, 則使用 `\ldots`。除了水平的連續點之外, 尚有垂直及對角方向之連續點:

\vdots `\vdots` \ddots `\ddots`

底下將舉例說明這兩個指令的用法。請注意, `\cdots`, `\vdots`, 與 `\ddots` 只能用於數式指令環境中; 但是 `\ldots` 可以用於數學環境, 也可以用於一般的文字排版中。

9.4 定義與定理

數學文稿常須排版定義、定理等, 而不同期刊書籍各有其排版定義與定理之格式。 \LaTeX 原提供 `\newtheorem` 指令, 讓使用者自行設定定義與定理的排版格式。Frank Mittelbach 寫了一套 `theorem` 巨集套件, 強化原有指令的功能。以上這兩套巨集指令雖然已有相當好的能力, 不過, 任何事情都有更上一層樓的空間。美國數學學會所發展的 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ 巨集套件讓排版數學文稿, 包括排版定義與定理, 更簡單, 也更精確、完美。

1980 年代初期開始發展時, 此巨集套件原稱為 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ 。其後, \LaTeX 系統逐漸普及。美國數學學會乃商請 Frank Mittelbach 與 Rainer Schöpf 重新發展為 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ 。 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ 含幾套巨集套件, 對於排版數學文稿而言, 其中最重要的可能是 `amsmath` 與 `amsthm` 巨集套件。 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ 是由美國數學學會所發展, 一般人可能認為它僅適用於複雜的數學文稿。其實, 即使是簡單的文稿也可從中獲得好處。譬如說, 數學式若想編號為 (3a), (3b) 等, 可輕易辦到, 但在 \LaTeX 中則難以解決。另外, \LaTeX 雖然提供 `eqnarray` 指令環境排版多行數學式, 但是 $=$ 號左右之空白卻顯得太大。若使用 `amsmath` 之指令, 結果會很完美。

`amsmath` 巨集套件之功能將於 9.6 節說明, 本節先介紹 `amsthm` 巨集套件。`amsthm` 的主要功能是排版數學文稿的定理與定義, 它是以 \LaTeX 之 `\newtheorem`

指令為基礎所發展出來的。此巨集套件雖然是 $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{E}\text{T}\text{E}\text{X}$ 之一部分，但也可直接用於一般的 $\text{E}\text{T}\text{E}\text{X}$ 文稿。

9.4.1 amsthm 巨集套件

數學文稿裡，定義、預備定理、或定理等，都自成一個段落，前端加上「定義」、「定理」等字眼以示區別。定義或定理可能加上編號，也可能不加。陳述定理之後，可能緊接著定理證明。簡單的定理證明可能只有一個段落，複雜的證明可能長達數頁。`amsthm` 直接提供 `proof` 指令環境以排版證明。在定理陳述部分，因為定理、引理、定義等種類較多，且格式不盡相同，故排版者須自行以 `\newtheorem` 定義指令環境。譬如，我們可自行定義 `defn` 指令環境以排版數學定義，或者定義 `thm` 指令環境以排版定理。

使用 `\newtheorem` 設定指令環境很容易。以定理為例，在全文設定區引入 `amsthm` 巨集套件之後，最簡單的格式如下：

```
\newtheorem{thm}{\ctxfbb 定理}
```

經過以上設定，我們即可使用 `thm` 指令環境排版定理，請見圖 9.1 之例子。

本例所排版的是中文稿，定理本身以明體字排版；在前面的設定，我們加入 `\ctxfbb` 中文字體指令，故「定理」兩字是粗黑體；但是，定理的編號是英文粗體字。圖 9.1 例子中，`\newtheorem` 指令之前有一行 `\theoremstyle` 指令，其作用是指定排版格式，包括英文字體的選擇。本例中，我們選擇 `plain` 格式。假設我們是排版英文稿，並作如下的設定：

```
\theoremstyle{plain}
\newtheorem{thm}{Theorem}
```

排版結果，定理陳述之前會加上粗體字 “**Theorem 1.**”，定理本身則以英文斜體字排版，這就是 `plain` 之格式。

除了 `plain` 格式之外，我們還可以選擇 `definition` 與 `remark` 兩種格式。三種格式中，`plain` 格式最醒目；`remark` 格式最不醒目，`definition` 則介於兩者之間。若使用 `remark` 格式，*Theorem 1.* 將以英文斜體排版，定理陳述本身則是標準的羅馬字體。如果對於以上三種格式都不滿意，我們還可以自行設計，請見底下之說明。

定理 1. 若 v 均等分佈於 $[a, b]$, 則 $E(v_{km}^{kn-1}) = a + (b - a)(n - m)/n$ 。

證明. 獨立抽取 $N - 1$ 個均等分配於 $U[0, 1]$ 的隨機變數, 令 z_M^{N-1} 代表其中第 M 大者。

...

由上式, $E(v_{km}^{kn-1}) = E(v_m^{n-1})$ 。 □

```

\usepackage{amsthm}
\theoremstyle{plain}
\newtheorem{thm}{\ctxfbb 定理}
\renewcommand{\proofname}{\ctxfms 證明}

\begin{thm}
若  $v$  均等分佈於  $[a, b]$ , 則 ...
\end{thm}
\begin{proof}
獨立抽取  $N-1$  個均等分配於  $U[0,1]$  的隨機變數,
...
由上式,  $E(v_{km}^{kn-1}) = E(v_m^{n-1})$  。
\end{proof}
    
```

圖 9.1: 定理與證明

利用同樣的方法, 我們可以設定排版定義 (definition), 預備定理 (lemma), 或其他陳述之指令環境。因為是自行定義, 使用者可以控制排版格式。譬如, 定理可以自動編上號碼, 也可以不編號。定理之編號數字可置於「定理」兩字左邊或右邊。

我們先說明排版證明之指令。使用 `proof` 指令環境時, \LaTeX 將於證明開始之前自動加上 *Proof.* 一字; 證明結束處則加上 \square 符號。本例為中文, 因此我們設定將 *Proof.* 改為中文。前 4 行之指令須輸入於全文設定區, 第 4 行指令設定將 *Proof.* 改為「證明」, 其原理是更改 `\proofname` 之定義。指令中 `\hspace*{\parindent}` 的作用是将該行內縮 `\parindent` 距離, 這是一般文字段落首行內縮的距離。經此設定, 證明文字之首行也內縮同樣距離。

依 `amsthm` 巨集套件的原始設定, *Proof* 一字是以斜體字排版, 其後並加上英文句點。本例中, 我們將句點改為冒號, 並選用正體字。遺憾的是, `amsthm` (1.2d 版) 並未提供更動此種設定之選項。因此, 我們須直接修改原始指令。在

amsthm.sty (1.2d 版) 檔案的倒數第 11 行末端有 `\itshape` 指令,其目的是選用斜體英數字形。去掉此指令即選用正體字。同檔案倒數第 10 行有 `\@addpunct{.}` 指令,將 `{.}` 改為 `{:}`,即改用冒號。請注意,為尊重作者版權,同時也為了避免產生混淆,作此更動之後,巨集套件檔案應另取新名,如 `mythm.sty`。

證明結束處之符號也可重新設定。例如,在全文設定區加入下列指令:

```
\newcommand{\qedsymbol}{\Box$}
```

符號將為 □,比原始正方形符號小一些。

接下來,我們說明排版證明陳述的方法,基本指令是 `\newtheorem`。此項指令可用於設定排版定義、定理、預備定理之格式。圖 9.1 的例子中,全文設定區有下列一行指令:

```
\newtheorem{thm}{\ctxfrb 定理}
```

此項指令設定排版定理的指令環境: `thm`。接下來,我們即可利用 `thm` 指令環境排版定理。排版時,定理陳述將自成一段落,與上下文之間的距離稍加大一些。定理陳述之前將自動加上「定理」兩字;自動編上號碼(以粗黑體排版);數字編號之後加上一英文句點。文稿中第二次使用 `thm` 指令環境時,定理號碼將編為 2。

若選擇自動編號,`\newtheorem` 指令另有兩個選項可以控制編號方法。以 `book` 文件類別為例,若第 3 章的定理要編為 3.1, 3.2 等,則定義 `thm` 指令環境之指令應改為:

```
\newtheorem{thm}{\ctxfrb 定理}[chapter]
```

指令最後面之 `[chapter]` 選項指示以章為編號單位。

另外一個情況,若預備定理與定理要混合編號,譬如,定理 2, 引理 3, 引理 4, 定理 5 等等,只要把下列指令加入圖 9.1 例子之全文設定區即可:

```
\newtheorem{lem}[thm]{\ctxfrb 引理}
```

以上指令定義 `lem` 指令環境,其中之 `[thm]` 選項表示新定義之 `lem` 指令環境將與 `thm` 指令環境混合編號。定理編號是附加於「定理」之後,如果在全文設定區加入 `\swapnumber` 指令,則編號將排於「定理」、「引理」等之前。如果不要編號,定義指令應改用 `\newtheorem*`。

<code>\newtheoremstyle{thmsty}%</code>	取名
<code>{3pt}%</code>	上方間距
<code>{3pt}%</code>	下方間距
<code>{}</code>	選用定理、定義陳述之英數字體
<code>{\parindent}%</code>	首行內縮距離
<code>{\sffamily}%</code>	選用標題之英數字體
<code>{:}%</code>	標題後加上冒號
<code>{.5em}%</code>	標題與定理陳述之間距
<code>{}</code>	特殊設定，請見巨集指令說明檔。

圖 9.2: 設計 amsthm 排版格式

有些定理或輔助定理有特別名稱, 如 Zorn's Lemma, 此時可以使用下列定義:

```
\newtheorem*{Zorn}{Zorn's Lemma}
```

除了排版證明之外, 我們也可以同樣方法設定排版定義、預備定理等之指令環境, 而且我們也可以選用不同的格式。amsthm 巨集套件提供三種格式: plain, definition, 與 remark。譬如, 在全文設定區輸入下列兩行指令:

```
\theoremstyle{definition}
\newtheorem{def}{\ctxfbb 定義}
```

文稿中即可使用 def 指令環境排版定義, 並選定 definition 格式。

萬一 amsthm 所提供的三種格式都不適合, 使用者還可以自行設計。設計排版格式須使用 `\newtheoremstyle` 指令。圖 9.2 是一個格式設計的例子。我們所更動的格式包括: 定理陳述首行內縮 `\parindent` 之距離; 「定理」兩字之後加上冒號。相對而言, 圖 9.1 中定理陳述首行並未內縮, 而「定理」兩字之後是加上英文句點。此外, 標題之英數字體以 `\sffamily` 字體族排版; 但定理陳述內容則以正體字編排。如果希望「定理」兩字要單獨排為一行, 定理陳述排於其下, 則倒數第 2 行大括號內應改填入 `\newline`。欲進一步了解 amsthm 巨集套件設計排版格式之指令, 請參考說明檔。

9.5 矩陣與行列式

數學文稿常排版矩陣或行列式, 矩陣的特徵是多項符號或字母規則性的排列, 並且左右各有圓括號或方括號。行列式與矩陣類似, 但無左右括號。行列式內各項

符號或文字之排列類似表格之排版, 介紹矩陣指令之前, 我們首先要了解如何排列變數或符號。下一章將介紹表格排版, 其中之 `tabular` 表格指令環境, 可使用於排列一般的文字, 但也可使用於排版數學符號。所以, 我們可以使用 `tabular` 指令環境排列出行列式或矩陣。不過, 排版行列式較方便的方法是使用數學表列 `array` 指令環境。

數學表列指令和排版表格之指令很類似, 請見底下的例子:

$a + b + c$	$m + n$	xy	<code>\[</code>
$a + b$	$p + n$	yz	<code>\begin{array}{clr}\</code>
$b + c$	mn	xyz	<code>a+b+c & m+n & xy \</code>
			<code>a+b & p+n & yz \</code>
			<code>b+c & mn & xyz</code>
			<code>\end{array}</code>
			<code>\]</code>

請注意, `array` 指令環境只能用於數學模式內。如同表格指令環境, 橫行各單項之間是以 `&` 區分, 每行之末則加上換行指令 `\\`。要注意的是, 最後一行的末端不須再加上換行指令。利用 `\begin{array}` 之選項, 可以控制每一欄內各數字或符號要居中、靠左或靠右排列。本例中, 純粹為了說明起見, 我們使用 `{clr}` 選項, 表示第一欄文字居中, 第二欄文字靠左, 第三欄文字靠右。

如果要排版矩陣呢? 很簡單, 只要在數學表列兩旁加上大括號即可。若為方括號, 指令為 `\left[` 與 `\right]`。此項指令所排版的括號, 其大小會因應數學表列大小自行調整:

$A = \begin{bmatrix} a + b & mn & xy \\ a + b & pn & yz \\ b + c & mp & xyz \end{bmatrix}$	<code>\[</code>
	<code>A = \left[</code>
	<code>\begin{array}{clr}</code>
	<code>a+b & mn & xy \</code>
	<code>a+b & pn & yz \</code>
	<code>b+c & mp & xyz</code>
	<code>\end{array} \right]</code>
	<code>\]</code>

第 6 章曾經說明, 版面上的每一行文字都有一條基線 (baseline)。當使用數學表列指令時, 全表之中央對準基線。本例中 'A =' 將對準表列之中間點。通常我們不會把基線對準中點外的其它地方。不過, 必要時 `array` 指令環境可以加上選項, 以決定基線的對應位置。字母 `t` 表示選擇表列頂點對齊基線; `b` 表示選擇底點; `c` 是內定值, 表示選擇中間點; 請見底下的例子:

表 9.13: 界限符號 (delimiter)

(({	\{	[\lfloor	↑	\uparrow
))	}	\}]	\rfloor	↓	\downarrow
[[<	\langle	↕	\Uparrow
]]		\	>	\rangle	↕	\Downarrow
[\lceil]	\rceil	/	/	↕	\updownarrow
				\	\backslash	↕	\Updownarrow

```

\left[
\Delta = \left(
\begin{array}{cc}
\alpha+b & \phi \\
\gamma & m\beta
\end{array}
\right)
\right]
    
```

上面兩個例子同時說明大括號的使用方法。在 \LaTeX 中，大括號或方括號稱為界限符號 (delimiter)。表 9.13 列出全部的界限符號。界限符號指令可以直接使用，也可以在其前面加上 \left 或 \right 指令。一旦加上 \left 或 \right 指令，界限符號的大小將隨其所包圍之數式或符號的大小自動調整。

要注意的是， \left 與 \right 指令原則上須成對出現。若僅輸入 \left ，忘了輸入 \right ，排版時將出現錯誤訊息。不過，以上指令所附加之界限符號可以左右不同。下例中，左邊為大括號，右邊為中括號。請注意，大括號之輸入指令為 $\text{\left\{}$ 。

```

\left\{
\Delta = \left\{
\begin{array}{c}
\alpha \\
\phi \\
\gamma
\end{array}
\right\}
\right]
    
```

以上所使用的界限符號 (包括左右括號) 都是由 \LaTeX 自行計算大小。若認為 \LaTeX 所決定之符號大小不適當，我們也可以自行選定大小。譬如，欲使用大除號，指令為 \big/ 。要大一些，指令為 \Big/ ；再更大一些，指令為 \bigg/ ；最大的是 \Bigg/ 。底下的例子說明符號之大小。

$$\} : \} \quad \text{\big}\} : \} \quad \text{\Big}\} : \} \quad \text{\bigg}\} : \} \quad \text{\Bigg}\} : \}$$

有些界限符號是左右成對出現的, 如左右圓括號。為了易為分辨, 以上之放大指令可加入 l 與 r。例如, `\bigl(` 與 `\bigr)`; 又如, `\Bigl[` 與 `\Bigr]`。

底下的例子較複雜, 但只要把段落分清楚, 各項指令並不難了解:

$$\left(\begin{array}{cc|cc} a & b & & \\ c & d & & A \\ B & & m & n \\ & & o & p \end{array} \right)$$

```

\l
\left(
  \begin{array}{cc}
    \left| \begin{array}{cc}
      a & b \\
      c & d
    \end{array} \right| & A \\
    B & \left| \begin{array}{cc}
      m & n \\
      o & p
    \end{array} \right|
  \end{array}
\right)
\l

```

界限符號原則上必須成對出現, 但有些數學式只用上左大括號。此時, 我們可以使用 `\right.` 指令代替右括號, 排版之後式子右邊將為空白:

$$x = \begin{cases} y & \text{if } y > 0, \\ z + y & \text{otherwise.} \end{cases}$$

```

\l
x = \left\{
  \begin{array}{ll}
    y & \mbox{if } \$y > 0$, \\
    z+y & \mbox{otherwise.}
  \end{array}
\right.
\l

```

本例中, 數學式內排版了英文字 “otherwise”。在數學式中排版一般文字時, 我們須以 `\mbox` 指令將文字圈入, 否則 \TeX 會將英文字當作數學符號排版, 變成 *otherwise*。同理, 數式內若要排版中文字, 也必須置於 `\mbox` 指令內, 例如:

$$x = \begin{cases} y & \text{若 } y > 0, \\ z + y & \text{其他情況。} \end{cases}$$

```

\l
x = \left\{
  \begin{array}{ll}
    y & \mbox{若 } y > 0, \\
    z+y & \mbox{其他情況。}
  \end{array}
\right.
\l

```

指令 `\mbox{}` 另外一個用途是在數式內加入一個不佔空間的虛擬字元, 其中左右大括號內不含任何字元。例如, `\mbox{}_tX_{t+1}` 可排版左下標符號: X_{t+1} 。若 `_t` 之前未加上 `\mbox{}`, 排版時將出現錯誤。

如果數學式子只需右括號, 則使用 `\left.` 替代左括號, 例如,

$$\left. \begin{aligned} \Gamma(z) &= \int_0^\infty e^{-t} t^{z-1} dt \\ \Gamma(z+1) &= z\Gamma(z) \end{aligned} \right\}$$

```
\[
\left.
\begin{array}{l}
\Gamma(z) \quad \& \int_0^\infty
e^{-t} t^{z-1} dt \\
\Gamma(z+1) \quad \& z\Gamma(z)
\end{array}
\right\}
\]
```

最後一個例子是以 `\ldots` 與 `\ddots` 指令排版矩陣:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

```
\[
A=\left(
\begin{array}{cccc}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \dots & a_{nn}
\end{array}
\right)
\]
```

9.6 amsmath 巨集套件

以上所舉的例子大多是單獨一行的數學式。那麼多行並列的數學式子如何編排呢? \TeX 特別提供了多行數式 `eqnarray` 指令環境。不過, 此一指令環境仍有一些限制。譬如, 上下兩行之數式僅能對齊其中一個等號, 若要對齊上下數式中之兩處, 即無法辦到。本節首先介紹 `eqnarray` 指令環境之使用方法, 接下來介紹功能較佳的 `amsmath` 巨集套件, 此為 \LaTeX 之一部分。

依排版規範, 若多行式子並排, 且每一行中各有一個等號 (或大於、小於符號) 時, 各行式子之等號應上下對齊。在 `eqnarray` 指令環境內, 上下對齊之符號是以兩個 `&` 符號圈於其中。下例中, 第一行數式之輸入指令較長, 我們以 `%` 指令折成兩行。第一行排版式子結束之處以 `\` 指令換行; 但最後一行之結尾不須加換行指令。

$$\begin{aligned} 1+r &= \frac{U_c(c_t, x_t)}{U_x(c_t, x_t)}, \\ c_t &= y_t + \frac{y_{t+1}}{1+r}. \quad (9.1) \end{aligned}$$

```
\begin{eqnarray}
1+r \quad \& \& \frac{U_c(c_t, x_t)}
{U_x(c_t, x_t)}, \quad \nonumber \\
c_t \quad \& \& y_t + \frac{y_{t+1}}{1+r}.
\end{eqnarray}
```

使用 `eqnarray` 指令環境時，每一行式子都自動編上號碼。若某一行不要編號碼，可以在其末端加上 `\nonumber` 指令。如果每一條式子都不要編號碼，可以直接使用 `eqnarray*` 指令環境。同一文稿內，`eqnarray` 與 `eqnarray*` 可以交互使用。

若以 `amsmath` 巨集套件排版類似的多行數式，可使用 `align` 指令環境：

$a_1 = b_1 + c_1 \quad (9.2)$ $a_2 = c_2 + e_2$	<pre>\begin{align} a_1 &= b_1+c_1\\ a_2 &= c_2+e_2 \notag \end{align}</pre>
-------------------------------------------------	-------------------------------------------------------------------------------------

首先，於全文設定區引入 `amsmath` 巨集套件。在 `align` 指令環境內，欲上下對齊等號，僅須使用一個 `&` 符號即可。另外，各式子會自動編號。欲取消編號，須在末端加上 `\notag` 指令。如果各式都不編號，應使用 `align*` 指令環境。

若數學式自動編號之格式不符需求，我們可以使用 `\tag` 指令以自選之格式編號。下例中，第 1 式編為 (3a)，第 2 式為 (3b)。若使用 `\tag` 指令， \TeX 自動加上左右括號；若使用 `\tag*` 指令，使用者須自行填入括號。

$a_1 = b_1 + c_1 \quad (3a)$ $a_2 = c_2 + e_2 \quad (3b)$	<pre>\begin{align} a_1 &= b_1+c_1\tag{3a}\\ a_2 &= c_2+e_2\tag*(3b)} \end{align}</pre>
-----------------------------------------------------------	------------------------------------------------------------------------------------------------

若數學式分為幾組，每一組在特定地方須上下對齊，我們仍可以 `align` 指令環境編排。底下的例子裡，每一行使用 3 個 `&` 符號，使兩組式子能在三個地方上下對齊。

$a_1 = b_1 \quad x_1 = y_1 \quad (9.3)$ $a_2 = c_2 \quad x_2 = y_2 \quad (9.4)$	<pre>\begin{align} a_1 &= b_1 &x_1 &= y_1\\ a_2 &= c_2 &x_2 &= y_2 \end{align}</pre>
---------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------

另一種情況是上下兩行或多行排列的數學式應視為一組，因此只需一個編號，此時我們可使用 `split` 指令環境。此指令環境之功能與下文介紹的 `multline` 指令環境相同；不過，拆開的各行仍可使用 `&` 指令以上下對齊。另外須注意者，`split` 指令環境並無自動編號功能。若要編號，可將此指令環境置於 `equation` 指令環境內，式子編號將出現於首行與末行中央。底下是一個簡單的例子：

$\begin{aligned} a_1 &= b_1 + c_1 \\ a_2 &= c_2 + e_2 \end{aligned} \quad (9.5)$	<pre>\begin{equation} \begin{split} a_1 &= b_1+c_1\\ a_2 &= c_2+e_2 \end{split} \end{equation}</pre>
----------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------

若上下式子並無對齊等號之必要, 可使用 `gather` 指令環境編排。排版之後, 每一式子將居中編排。

$a_1 + b_1 + c_1 \quad (9.6)$ $a_2 + b_2 + c_2 + d_2 \quad (9.7)$	<pre>\begin{gather} a_1 + b_1 + c_1 \\ a_2 + b_2 + c_2 + d_2 \end{gather}</pre>
-------------------------------------------------------------------	---------------------------------------------------------------------------------

最後, 若一個式子太長, 無法編排在一行之內, 我們可使用 `multline` 指令環境編排, 在欲換行處加上 `\\` 換行指令即可。一長式子若排為 3 行 (需兩個換行指令), 第 1 行靠左編排, 第 3 行靠右編排, 第 2 行則居中編排。各行之間距可由 `\multlinegap` 控制, 我們可以使用 `\setlength` 或 `\addtolength` 設定其值。

排版多行並列數式, 或者是把一條長式子拆成幾行時, 爲了提高可讀性, 我們必須調整各行式子的相對位置, 底下是一個參考例子:

$\begin{aligned} \sum_{i=1}^{\infty} x_i + \sum_{i=1}^{\infty} y_i &= \\ x_1 + x_2 + x_3 + \dots + & \\ y_1 + y_2 + y_3 + \dots & \end{aligned}$	<pre>\usepackage{amsmath} \begin{align*} \lefteqn{\sum_{i=1}^{\infty} x_i + \sum_{i=1}^{\infty} y_i} &= \\ & x_1 + x_2 + x_3 + \\ & \quad \dots + \\ & y_1 + y_2 + y_3 + \\ & \quad \dots \end{align*}</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

此例中, 我們使用 `\lefteqn` 指令將第一行式子全部括於其中。排版時大括號內全部式子將視爲一符號, 且寬度爲零。換言之, 第一行式子不具有任何寬度, 而第 2 行與第 3 行的兩個 `&` 符號所包含之空白將上下對齊。因爲每一行的兩個 `&` 符號本身會產生一點小空白, 因此版面上第 2 行與第 3 行將往右移動一點距離, 整段式子看來較清楚。

若有一很長式子拆成兩三行, 而且第二或第三行是以加減號起頭, 則等號不宜直接與加減號對齊。簡單的解決辦法是在第二行加入 `\quad`, 讓第一行之等號對齊第二行開頭處之空白:

$$y = x_1 + x_2 + x_3 \\ + x_4 + x_5$$

```
\usepackage{amsmath}
\begin{align*}
y &= x_1 + x_2 + x_3 \\
&\quad + x_4 + x_5
\end{align*}
```

本例是以 align* 指令環境編排, 因此數式並不編上號碼。

9.7 細節調整與數式編號

在數學環境下, \TeX 會自動處理許多排版上的細節, 例如數學符號以斜體字編排, 符號之間的距離適當調整, 上下標選用較小字體, 數學式自動編上號碼等等。但是, 還是有某些情況, \TeX 的處理結果不盡理想, 必須進一步人為調整。以下分別說明之。

9.7.1 調整符號間距

在數學環境中, 各符號之間距會自動調整。不過, \TeX 終究只是電腦軟體, 在特殊的情況下, 它所設定的結果可能不理想。此時, 我們須作進一步調整。底下列出數學模式內調整間距的指令:

- \backslash , 加入小空白 (約 1.5pt)
- $\backslash:$ 加入中等空白 (約 3pt)
- $\backslash!$ 減去小空白 (約 1.5pt)
- $\backslash;$ 加入大空白 (約 5pt)

指令 $\backslash!$ 可視為是 \backslash 的相反。前者縮小空白, 後者增加空白。底下的例子中, 左邊是排版結果; 中間是輸入指令, 其中已加入調整空白之指令; 右邊是不加上調整指令時的排版結果。

$\sqrt{2}x$	$\backslash\sqrt{2}\backslash, x\$$	$\sqrt{2}x$
$n/\log n$	$\backslashn/ \backslash! \backslash \log n\$$	$n/\log n$
$\iint z dx dy$	$\backslash\int\int\backslash!\backslash\int z\backslash, dx\backslash, dy\$$	$\int \int z dx dy$
$x^2/3$	$\backslashx^2\backslash!/3\$$	$x^2/3$

通常, 我們要看到初步的排版結果之後, 才能決定那些地方須作調整。因此, 剛開始輸入時, 你不須擔心是否要加上調整指令。等到已屆完稿階段, 才作間距調整。

9.7.2 調整符號大小

依據排版規範，數學式之上下標符號須縮小一些；上下列分式之分子分母符號亦然。TeX 會儘量將每一個數學符號調整至正確大小。但是，人為調整仍不可免，連分式 (continued fraction) 是一個有名的例子。在底下的例子中，我們須使用三個 `\displaystyle` 指令強制分母以較大之字體編排。

$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}}$	$\backslash[$
	$a_0 + \frac{1}{\displaystyle a_1$
	$+ \frac{1}{\displaystyle a_2$
	$+ \frac{1}{\displaystyle a_3$
	$+ \frac{1}{a_4}\}\}$
	$\backslash]$

若不加 `\displaystyle` 指令，TeX 以內定之大小排版分母，各分母項之字體會越來越小，結果並不理想：

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}}$$

本例中，我們使用 `\displaystyle` 控制數學符號大小；全部的控制指令計有底下四個：

<code>\displaystyle</code>	展示數學符號之標準大小，
<code>\textstyle</code>	隨文數學符號之標準大小，
<code>\scriptstyle</code>	第一層上下標之標準大小，
<code>\scriptscriptstyle</code>	第二層以後上下標之標準大小。

9.7.3 數學式居中與靠左

依原始設計，數學式將居中排版。若希望靠左編排，應於 `\documentclass` 指令中加入 `fleqn` 選項。例如：

```
\documentclass[11pt,fleqn]{article}
```

靠左之數學式若要靠版面左緣一點距離，可在全文設定區以下列指令設定：

```
\mathindent=1cm
```

即設定每一展示數式自左沿內縮 1 公分。

9.7.4 交叉引述

數學式會自動編上阿拉伯數字號碼。若使用 `chapter` 文件類別排版, 在第 9 章內, 數學式將依序編號為 (9.1), (9.2), ...。文稿其他地方若要引述某數學式, 可以使用引述指令: `\label` 與 `\ref`。

以下的例子中, 數式之編號是以 `\tag` 指令自行加上。被引述之數學式須以 `\label` 指令標示, 本例中, 兩行數學式分別標示為 `xy` 與 `ab`:

$z = x + y,$	(2a)	<code>\usepackage{amsmath}</code>
$c = a + b.$	(2b)	<code>\begin{align}</code>
		<code>z &= x+y, \tag{2a} \label{xy} \\</code>
		<code>c &= a+b. \tag{2b} \label{ab}</code>
		<code>\end{align}</code>

經過標示之後, 文章他處即可以 `\ref{xy}` 指令引述式 (2a)。同理, `\ref{ab}` 即可徵引式 (2b)。具體言之, 文稿內若鍵入「...數式 (`\ref{ab}`) ...」, 連續執行 `latex` 兩次之後, 排版結果將變成「...數式 (2b) ...」。換言之, \LaTeX 將 `ab` 標誌, 替代文稿之數式編號 2b。

如果排版文稿須反覆修改, 則使用 `\label` 與 `\ref` 指令甚為方便。譬如, 在初稿中數學式之編號為 (2a), (2b); 而修正稿中已變成 (4a), (4b)。此時, 以 `\ref` 指令徵引數式之處會自動更新編號。請注意, 使用 `\ref` 指令時, 我們須連續執行 `latex` 兩次, 才能產生正確結果。

數式編號是以 `equation` 計數器為之。假設文稿一開始的 4 條數式由 \LaTeX 自動編號, 接著的兩條式子以 `\tag` 指令自行編號為 (10.5a) 與 (10.5b), 再下一條數式應為 (10.6)。此時, 我們可以必須先調整數式之計數器, 才能得到正確結果:

```
\setcounter{equation}{5}
```


10 表格

若安排得當，一個簡單的表格勝過千言萬語。製作表格的目的是使讀者能夠迅速地掌握資料或訊息。因此，表格應該簡單、乾淨、準確。若表格太複雜，讀者無法很快地了解其內容，反而失去以表格表現資訊的目的。對於初學者而言，排版表格可能是 \LaTeX 最困難的一部分。不過，如果能善用巨集套件，排版表格並不困難，而且排版品質是一般文書處理軟體遠不能及的。

在 \LaTeX 中，排版表格可以使用 `tabbing`，`array` 與 `tabular` 指令環境。若以上指令環境還不能解決問題， \LaTeX 另外提供 `array` 巨集套件，目的是強化 `array` 與 `tabular` 指令環境之功能。除此之外，我們尚可使用 `tabularx` 與 `dcolumn` 等巨集套件排版表格。除了以上巨集套件之外，本章也將介紹如何排版彩色/灰階表格，以及如何處理超大型表格。

若熟悉本章介紹的巨集套件，即使是複雜表格也不難處理，但這並不表示我們一定能排版出「高排版品質表格」。事實上，從排版的角度來說，困難的問題不是「如何排版」，而是了解什麼是「品質」。一般表格排版最常見的錯誤是加入太多不必要的橫線或直線。因此，在介紹表格指令之前，我們先簡單歸納專家眼中表格排版的要點。

根據專家的意見，排版表格應注意下列要點：

- 表格中絕勿畫垂直線，
- 不要畫兩條緊鄰橫線，
- 數字單位應排於欄位上端，而非欄位內，
- 小數點之前應加上 0，例如 .5 應排為 0.5，
- 本欄位數字與上一欄位相同時，請勿使用「同上」；應直接排出數字。

以上意見引自 `booktabs` 巨集套件，作者是 Simon Fear。你或許覺得這些意見極端，不過，研究形形色色的表格之後，你會發現以上各點都很有道理。舉例來說，

底下所排版的這兩個表格各有三欄資料，左邊表格各欄之間是以直線區分，右邊表格則拿掉各欄之間的直線，上線與底線較粗，中間線較細。右邊表格所含資料並不減少，但版面清爽、易讀。

國家	央行 獨立性	物價 上漲率 (%)
意大利	0.5	16.1
英國	2	12.3
加拿大	2	8.1
西德	4	4.1

國家	央行 獨立性	物價 上漲率 (%)
意大利	0.5	16.1
英國	2.0	12.3
加拿大	2.0	8.1
西德	4.0	4.1

因此，排版之前，我們應該先思考何種設計能提高表格的易讀性。如果表格複雜到一般的 \LaTeX 指令都難以處理，則與其絞盡腦汁思考排版方法，倒不如想一想是否有更容易表達想法的列表方式。換言之，表格排版的目的是希望精簡地傳達重要訊息，如果表格的設計複雜到不易排版，那表示讀者可能難以掌握作者所欲傳達的訊息。此時，簡化表格設計是第一步應該作的事情。

以上例子中，表格置於行文當中。事實上，在專業排版中，大表格通常不置於文章中間，而是移放版面上方或下方。爲了此一目的， \LaTeX 提供 `table` 與 `figure` (浮動版面) 指令環境，其主要功能是自動尋找適當置放圖表之位置。在此指令環境內，我們可使用 `\caption` 指令排版圖表標題， \LaTeX 會自動編入圖表號碼。

排版表格主要是使用 `tabbing` 與 `tabular` 指令環境。在介紹指令之前，我們先簡單說明兩者之差別。指令環境 `tabbing` 的主要功能是將文字/數字排列於版面適當位置，它並無現成指令可供加入橫線或直線。欲排版具有橫線或直線的表格，最好是使用 `tabular` 指令環境。不過，`tabbing` 指令環境所排版之表格可以跨越一頁以上，`tabular` 則不行。若表格中有橫線條，而資料又多到無法擠進一頁當中，我們可以使用 `longtable` 巨集套件，其指令功能與 `tabular` 類似，但可處理超長或超寬表格。

如果表格複雜，`tabular` 指令環境內可以包含另一個 `tabular` 指令環境。因此，我們可以用它來排版一個大表格，其內包含幾個小表格。相對的，`tabbing` 指令環境中則不能包括另一個 `tabbing` 指令環境。一般而言，排版表格大多使用 `tabular` 指令環境；不過下一節我們先介紹 `tabbing` 指令環境，再下一節則介紹 `tabular` 指令環境之各種應用。

	大	中	小
牛肉麵	120 元	100 元	80 元
搾醬麵	60 元	50 元	40 元
酸辣湯	40 元	30 元	20 元

```

\begin{tabbing}
\ctxfk
\hspace*{1.1cm}\=~~ 大\qqad\= %
~~中\qqad\=~~ 小\\
\ctxfm
牛肉麵 \> 120 元 \> 100 元 \>80 元\\
搾醬麵 \>~~60 元 \>~~50 元 \>40 元\\
酸辣湯 \>~~40 元 \>~~30 元 \>20 元
\end{tabbing}

```

圖 10.1: 以 `tabbing` 指令環境排版表格

10.1 `tabbing` 指令環境

排版表格的方法之一是使用 `tabbing` 指令環境，此一指令環境和傳統打字機編排表格的原理類似。傳統打字機鍵盤，左上方有一 [Tab] 鍵，按下此鍵，打字頭將往右方移動若干距離。TeX 的 `tabbing` 指令環境即模倣 [Tab] 鍵之功能。如果表格內容主要是一些上下對齊的文字符號，其間沒有橫線或直線，`tabbing` 指令環境很適合用來排版。反之，如果表格中有區隔之直線或橫線條，使用 `tabular` 指令環境排版比較容易。

圖 10.1 之例子表格共有 4 欄，各欄之間距須自行設定。本欄之數字或文字排版後，要跳至下一個欄位須使用 `\>` 指令，此項指令之功能與打字機上之 [Tab] 鍵類似。本例中，`tabbing` 指令環境內含數行指令，第一行除了排版標題之外，並且使用 `\=` 指令設定欄寬；行末之 `\\` 表示此行指令結束。

表格內之文字如何排列？設想表格中每一行最前端隱藏第 0 個 [Tab]；第 0 到第 1 個 [Tab] 之間即容納第 1 欄文字，第 1 到第 2 個 [Tab] 之間即容納第 2 欄文字。圖 10.1 例子中，第 1 欄內之「牛肉麵」、「搾醬麵」等文字即在第 0 個 [Tab] 位置排出；第 2 欄之價格數字即排於第 1 個 [Tab] 位置。此例中，有些價格是三位數，有些是二位數；排版時最好是個位數上下對齊。在 `tabbing` 指令環境中，文字/數字原本是緊貼著 [Tab] 位置排出。為了使個位數上下對齊，輸入價格數字時，我們特別在 60 或 40 等數字之前加上兩個調整空白之指令 `~~`。

欄位寬度如何控制？本例中，輸入指令的 3-4 行含有控制欄寬之設定。第一欄是以 `\hspace*{1.3cm}` 指令設定欄寬為 1.3cm，其後接之 `\=` 指令即標示第 1 個 [Tab] 位置。接著排版「大」字，其後以 `\qqad` 指令拉開一點空白；接著是第 2 個 [Tab] 之位置。「大」字之前另外加上兩個 `~~` 指令，以免標題太靠欄位左邊。

股市	收盤	漲跌幅
美國道瓊	10913.3	-1.1
法國	4323.8	0.8
台灣	7576.6	1.4

```

\begin{tabbing} \ctxfk
美國道瓊數\= 222222222\=\kill
~~股市 \> ~~收盤 \> 漲跌幅\\
\ctxfm
美國道瓊 \> 10913.3\> $-1.1\\
法國 \>~~4323.8\> ~~0.8\\
台灣 \>~~7576.6\> ~~1.4
\end{tabbing}

```

圖 10.2: 以樣本行設定欄寬

第 2、3 欄之寬度也是以類似的指令設定。第 2 行開始，鍵入「牛肉麵」三個字之後，使用 \> 指令即跳至第一個 [Tab] 位置。若某欄位空白，連續兩個 \> 指令即可跳到再下一個 [Tab] 位置。

本例中，每一行的第 4 欄都有文字。若某一行的第 4 欄空白，則該行第 3 欄文字之後直接加上換行指令 \\ 即可結束該行。另外，最後一行尾端不須加上換行指令，因為 \end{tabbing} 指令即兼有結束最後一行之功能。

10.1.1 以樣本行設定距離

圖 10.1 的表格例子中，設定欄寬的第一行文字是表格的一部分，會在排版結果中出現。另外一種方式是以一樣本行 (sample line) 設定欄寬，排版時樣本行並不出現，請見圖 10.2 的例子。指令第一行為樣本行，行末須以 \kill 指令作為結束，不能使用 \\ 指令；最右欄之欄寬不須設定。因為樣本行之文字不會排版出來，我們可以鍵入任何文字或以距離指令設定 [Tab] 位置。譬如，第 1 欄寬度是以「美國道瓊數」文字設定距離；第 2 欄則以 9 個「2」字設定距離。這些文字/數字與表格內容並無任何關係。

本例中，美國道瓊股票指數之漲跌幅度為 -1.1。請注意，排版減號應使用數學模式：\$-1.1，或者 \$-1.1\$。如果未進入數學模式，直接鍵入指令 -1.1；排版結果為 -1.1；代表減號之橫線顯得太短。此外，為了使各數字小數點第一位上下對齊，數字前端加上數個控制空白之指令；但結果仍不理想，下文會介紹使上下數字對齊之方法。

在每一欄位中，文字將從指定之 [Tab] 位置開始排版。換言之，文字串的最左端將位於 [Tab] 位置。但我們也可以設定將文字串之右端擺於 [Tab] 位置，此時須在文字串尾端加上 \' 指令。若此一指令是加於一串文字中間，指令左邊的

年期	館藏地
大正11年	台灣分館
12	台大總圖
昭和元年	台大總圖
2	台大總圖

```

\tabbingsep=0.2mm
\begin{tabbing}
\hspace*{1cm}\= 22222222\=\kill
\ctxfk
\>年\’ 期 \> 館藏地 \\
\ctxfm
\>大正\’11年 \> 台灣分館\\
\> 12 \> 台大總圖\\
\>昭和\’ 元年 \> 台大總圖\\
\> 2 \> 台大總圖
\end{tabbing}

```

圖 10.3: tabbing 指令環境之應用

文字將排於 [Tab] 位置之左, 其右再接著排出文字串右邊文字。利用此一指令, 我們可設定將上下數行之文字對齊某一點。

圖 10.3 是一個應用的例子。此例中, 我們設定兩個 [Tab], 第 1 個 [Tab] 位置距離版面左邊緣 1 公分。利用 \’ 指令, 我們將「大正」、「昭和」等文字挪於 [Tab] 左方, 右方則排版「11 年」、「元年」等文字。接下來的「台灣分館」、「台大總圖」等將從第 2 個 [Tab] 位置開始編排。第 1 個 [Tab] 左右兩邊文字之間隔可以用 \tabbingsep 指令控制。此外, 我們利用 \tabbingsep 指令將間距設定為 0.2mm。

與 \’ 指令相對的是 \’, 其功能是将排版文字往右推擠到版面邊緣。如果某段文字之前加上 \’ 指令, 而且其後沒有 \= 或者 \> 指令, 則此段文字將居右 (right-justified) 排版。

以上所介紹的 \=, \’, 與 \’ 等三個指令, 原本是用以排版重音符號 (參見表 5.1 節, 頁 66)。在 tabbing 指令環境中, 因為它們各有特別作用, 因此喪失其原來的功能。如果在 tabbing 指令環境中要排版重音符號, 我們須分別以下列指令替代: \a=, \a’, 與 \a’。例如, 要排版 ö, 我們須鍵入 \a=o。

10.1.2 其他控制指令

在 tabbing 指令環境下, 若一行之首加上 \> 指令, 第 1 欄文字將從第 1 個 [Tab] 位置開始編排, 而非隱藏的第 0 個 [Tab]。若表格共有 5 行, 每一行之首都須加上 \> 指令。為了簡化輸入, 我們可以在第一行之首以 \+ 指令替代 \> 指令, 以下各行前端就不須再加上任何指令, 各行最左欄文字/數字就會從第 1 個 [Tab] 位置

開始編排。

如果某一行之首加上 `\+{+}` 指令, 則從該行開始以下的每一行文字將從第 2 個 [Tab] 位置開始編排。相反的, `\-` 指令則使下一行開始編排之欄位往左移一個 [Tab] 位置。最後, 若某一行之首加上 `\<` 指令, 則前面所加之 `\+` 指令在該行即失效。

10.2 array 巨集套件

除了 `tabbing` 指令環境之外, 排版表格尚可使用 `tabular` 與 `array` 指令環境; 其中, `array` 指令環境主要是用於排版數學行列式與距陣, 請見 9.5 節之說明。因為表格經常出現在文稿中, 而且式樣多變, 新版 \TeX 中另外提供 `array` 巨集套件以加強 `tabular` 指令環境之功能。

因為表格樣式變化甚多, 因此除了以上 \TeX 所提供的指令, 許多專家又寫了各種巨集套件。譬如, Simon Fear 設計 `booktabs` 巨集套件, 以配合 \TeX 之 `tabular` 指令環境使用。本節主要介紹 `tabular` 指令環境與 `array` 巨集套件之強化功能, 稍後並介紹 `booktabs` 巨集套件。

10.2.1 tabular 指令環境

使用 `tabular` 指令環境排版表格並不難, 但是, 要控制到表格的每一個細節, 我們需仔細了解各項指令。學習排版表格最好的方法是多研究一些例子, 因此底下提供眾多例子說明。

圖 10.4 是 `tabular` 指令環境的第一個例子, 控制格式之指令如下:

```
\begin{tabular}{lccc}
```

其中, `{lccc}` 選項用以設定各欄位文字/數字之排版位置。括號內 4 個字母表示表格共有 4 欄資料, 第一個字母 `l` 代表 `left` (左邊), 表示第一欄資料應靠左排列。2-4 個字母都是 `c` 代表 `center` (居中), 表示 2-4 欄資料排版時應居中。如果 2-4 欄資料靠右編排, 指令應改為 `\begin{tabular}{lrrr}`, 其中字母 `r` 代表 `right`。

表格資料如何輸入呢? 每一橫行的 4 筆資料之間是以 `&` 符號分間, 行末則加上換行指令 `\\`。若要調整本行與下一行之行距, 可以在換行指令之後設定。本例

國家	央行 獨立性	物價 上漲率	支出 比率
意大利	0.5	16.1	35.6
英國	2	12.3	28.4
加拿大	2	8.1	23.1
西德	4	4.1	29.3

```

\begin{tabular}{lccc}
\hline
& 央行& 物價& 支出 \\
& 國家 & 獨立性 & 上漲率 & 比率 \\
\hline
意大利 & 0.5 & 16.1 & 35.6 \\
英國 & 2 & 12.3 & 28.4 \\
加拿大 & 2 & 8.1 & 23.1 \\
西德 & 4 & 4.1 & 29.3 \\
\hline
\end{tabular}

```

圖 10.4: tabular 指令環境

中第一行標題尾端的指令為 `\[-2pt]`，目的是將兩行標題之行距減小一些，以求標題更清楚。若某一欄資料從缺，該欄位就留為空白，不須輸入任何資料。

排版時，每一欄的寬度如何決定呢？在 `tabular` 指令環境中，欄寬可以自行決定，也可以讓 \TeX 計算決定。本例中，欄寬是由 \TeX 計算決定，計算方法如下： \TeX 檢查某一欄下各項資料排版之後的寬度。選擇其中最寬者，左右再加上適當空白，結果就是該欄的寬度。本例中我們加入 3 條水平線，畫水平線的指令為 `\hline`。連續兩道 `\hline` 指令將排版出兩條緊鄰橫線。不過，請記住專家的意見：表格中避免畫兩條緊接著的橫線。

排版時， \TeX 將整個表格視為一個字元。因此若整套表格指令是緊接在一段文字之後，表格會出現在行文中間，表格中央將對齊緊接其前之文字。如果希望表格上端對齊前接文字，應加上 `[t]` 選項：

```
\begin{tabular}[t]{lrrr}
```

相反的，若表格下沿要對齊前接之文字，設定項應為 `[b]`。把表格當成是一大字母來處理有一個好處：排版時，表格不會被拆開為兩部分，上半部分在本頁底下，下半部分在下一頁的開頭。如果表格要置於橫跨版面的中央，僅須將 `tabular` 指令環境置於 `center` 指令環境內即可。因為表格是視為一個大字母，故另一個作法是在 `tabular` 指令環境之前加上 `\centering` 指令。

因為整個表格當成是一個字母處理，因此要將兩個表格並排很容易，只要把兩個 `tabular` 指令環境緊接著鍵入即可。請注意，兩個指令環境之間不可留下空格，否則表格可能變成下上排列，而非平行並列。必要時，還可以用 `\hspace` 指令

控制兩個表格之間距。譬如, 本章開頭之並排表格即以下列指令排版:

```
\begin{tabular}{|l|c|c|}
... [左邊表格之指令]
\end{tabular} \hspace{.05\textwidth}
\begin{tabular}{lcc}
... [右邊表格之指令]
\end{tabular}
```

我們使用 `\hspace` 指令將兩個表格之間隔定為版面寬度的 5%。仔細觀察圖 10.4, 我們發現欄位標題第一行文字與其上橫線之距離稍嫌小了一些。同樣的, 表格內第一橫行文字與其上橫線之間距也稍嫌小了一些。主要原因是表內含有中文字。一般而言, 中文字較英數文字為高。若利用現成的表格格式排版表格, 通常須作些許調整。

欲拉大表格內每一橫行之間距, 可以使用 `\extrarowheight` 指令。譬如, 若在 `tabular` 指令環境之前加上:

```
\extrarowheight=2pt
```

即可將行距加大 2pt。

圖 10.4 的表格並未加上任何垂直線。排版專家的意見是表格中不應該加入任何垂直線。如果非加上垂直線不可, 只須在 `tabular` 指令環境的設定項中加上代表直線的指令 `|` 即可。舉例來說, 若設定項變成:

```
\begin{tabular}{|l|r|r|r|}
```

如果是連續下兩個直線指令 `||`, 排版後即出現兩條緊鄰的垂直線。排版之後, 各欄資料之前後將會出現一直線, 如圖 10.5 所示。請注意, 我們已將 2-4 欄文字改成居右排版, 表格上端只畫一橫線, 而且行高已加大 2pt。另外, 標題文字之後我們輸入兩道 `\hline` 指令, 畫出兩條水平直線。

10.2.2 控制欄位間距

表格中各欄位間之空隔距離內設值為 12pt, 但此一間距可以更改之。要將間距減為 6pt, 僅須在 `tabular` 指令環境之前加上下列指令:

```
\tabcolsep=3pt
```


國家	央行 獨立性	物價 上漲率	支出 比率
意大利	0.5	16.1	35.6
英國	2	12.3	28.4
加拿大	2	8.1	23.1
西德	4	4.1	29.3

圖 10.5: 表格內加上直線

換言之, `\tabcolsep` 指令設定欄位間距之一半值。

以上之欄距指令同時更動所有欄位之間距。如果我們只想要更動某兩欄位之間距, 可以使用 `@{...}` 指令。譬如, 圖 10.4 第 2 欄與第 3 欄之間距若想設定為 0.1 公分, 指令為:

```
\begin{tabular}{lc@{\hspace{1mm}}cc}
```

在此設定下, 第 2 欄位與第 3 欄原有之間距被取消, 代之以選定之距離。事實上, `@{...}` 指令大括號內除了設定欄位間距外, 也可以鍵入任何文字或指令。排版時, 括號內之文字或宣告指令即自動填入表格中對應的欄位間隔處, 原有之空白自動消除。底下舉一個例子說明其用途。

在專業排版中, 表格內的數字若有小數點, 排版時小數點應上下對齊。就此點而言, 本章一開頭之表格並不符合標準。若使用 `@{...}` 指令, 原來表格可重新排版如圖 10.6。本例中, 排版之後有 3 欄, 但第 3 欄之數字爲了特別處理小數點上下對齊的問題, 整數與小數部份各占用一欄排版。設定指令 `@{.}` 的作用是在表格的對應欄位處排版小數點。左欄使用 `r` 設定指令, 用以排版整數; 右欄使用 `l` 指令, 用以排版小數。因此兩欄位所鍵入之整數及小數, 連同小數點即組成正確的數字。

要注意的是, 如果輸入時我們鍵入 `@{.}`, 亦即小數點前後各留下一空白, 排版之後小數點前後也會出現空白。因此, `@{...}` 指令的功能是先去掉原來內設之任何空白, 再將大括號中任何文字或指令原封不動地排版出來。以上的方法雖然可以得到正確的結果, 但稍嫌麻煩。第 10.6.1 節將介紹 `dcolumn` 巨集套件, 可以較容易地解決小數點對齊的問題。

意大利	0.5	16.12
英國	2	12.3
加拿大	2	8.1
西德	4	4.15

```

\begin{tabular}{rlr@{.}l}
\hline
意大利 & 0.5 & 16& 12 \\
英國 & 2 & 12& 3 \\
加拿大 & 2 & 8& 1 \\
西德 & 4 & 4& 15 \\
\hline
\end{tabular}

```

圖 10.6: 表格數字上下對齊

10.2.3 booktabs 巨集套件

利用 `tabular` 指令環境內之 `\hline` 指令所畫之表格橫線，粗細都相同。若要畫不同粗細之橫線，可以使用 `booktabs` 巨集套件。巨集指令之作者 Simon Fear 為一專業排版者，利用此巨集套件排版之表格，結果相當優美，值得推薦使用。

在 `booktabs` 巨集套件下，我們可以使用下列橫線指令：

```

\toprule[ wd]
\midrule[ wd]
\bottomrule[ wd]
\cmidrule[ wd](trim){a-b}

```

前三行指令分別用於畫表格上方、中間及底下之橫線；`[wd]` 選項用於設定線條之粗細。若上方橫線之粗細要定為 1pt，指令為 `\toprule[1pt]`。若不加選項，程式會自動選用較粗之線條畫 `\toprule` 與 `\bottomrule`。

圖 10.7 之表格有三欄資料，我們使用 `\toprule` 等指令畫三條粗細不同之橫線，其長度與表格寬度相同。請注意，橫線上下方之空白會自動加大，不須再調整。中間的短橫線只貫穿兩個欄位，我們使用 `\cmidrule(1){2-3}` 畫出。`{2-3}` 設定短線橫跨第 2–3 欄位；圓括號 (1) 選項表示橫線左方要切短一點。若短線左右端都要截短，指令為 `\cmidrule(1r){2-3}`。

若連續使用兩個 `\cmidrule` 指令，排版結果兩條短線會接在一起，無法分辨。為解決此問題，可將左邊短線之右端及右邊短線之左端各切下一小截。若表格共有 5 欄，2–3 欄及 4–5 欄上方各要畫一短線，指令為：

```

\cmidrule(r){2-3}\cmidrule(1){4-5}

```

請注意，兩道指令之間不可留空格，否則兩條線無法水平對齊。同一欄位上若設

經濟表現		
國家	物價 上漲率	政府支出 比率
意大利	16.1	35.6
英國	12.3	28.4
加拿大	8.1	23.1
西德	4.1	29.3

```

\usepackage{booktabs}
\begin{tabular}{@{}lrr@{}}
\toprule
& \multicolumn{2}{c}{ 經濟表現} \\
\cmidrule(1){2-3}
& 物價 & 政府支出 \\
& 上漲率 & 比率 \\
\midrule
意大利 & 16.1 & 35.6 \\
英國 & 12.3 & 28.4 \\
加拿大 & 8.1 & 23.1 \\
西德 & 4.1 & 29.3 \\
\bottomrule
\end{tabular}

```

圖 10.7: booktabs 巨集套件

定兩道指令，將排出緊接之兩條短線，例如：

```
\cmidrule(1){2-3} \cmidrule(1){2-3}
```

2-3 欄短橫線上方的「經濟表現」四個字占用兩個欄位。欲排版占用兩欄位以上之文字標題，可使用下列指令：

```
\multicolumn{ n}{col}{text}
```

其中，*n* 代表占用欄位數；*col* 設定要居中、靠右或靠左。本例中，文字占用兩欄，並居中排版。

圖 10.7 之表格例子中，欄位指令選項之前後端各加上一個 @{} 指令，其目的是除掉多餘的空白。這是專業人士建議的作法。若不加上此設定，表格內容之寬度將比橫線小一些。本例中，\cmidrule 是 booktabs 巨集套件特別提供的指令。如果直接使用 tabular 指令環境排版，指令為 \cline{2-3}。請注意，\cline 指令之後僅能設定橫跨欄位數，不能控制線條粗細，也無法像 \cmidrule 指令設定 trim 之功能。如果我們以 \hline 替代 \toprule 等橫線指令，以 \cline{2-3} 替代 \cmidrule 指令重新排版，結果如圖 10.8 所示。

10.2.4 控制中文字距

以上的表格例子並未加上標題。欲在圖表加上標題，可使用 \caption 指令。不

國家	經濟表現	
	物價 上漲率	政府支出 比率
意大利	16.1	35.6
英國	12.3	28.4
加拿大	8.1	23.1
西德	4.1	29.3

圖 10.8: 表格內短橫線

過, 此一指令僅能下於浮動版面指令環境內, 我們將在稍後說明使用的方法。底下先說明自行排版標題的方法。圖 10.9 的例子爲了使標題各中文字之間距稍微拉開一些, 我們特別定義一巨集指令 `\csp` 如下:

```
\newcommand{\csp}[2]{\let\zori=\z \let\z=\hfill
\makebox[#1]{#2}\let\z=\zori}
```

請注意, 巨集指令必須定義於全文設定區。此外, 此巨集指令僅能用於中文字, 英文字串無效。

巨集指令 `\csp` 之原理如下: `cwtext` 處理中文時, 會在兩個中文字中間加上指令 `\z`, 目的是在控制中文字之間距。而 `\csp` 巨集指令將 `\z` 重新定義爲一伸縮彈性無窮大之彈簧。因此, 當我們限制標題寬度之後, 各中文字之間距會盡量加大, 產生均勻空白之效果。使用巨集指令時須先填入標題長度, 之後再填入文字。

同學會通訊錄	
姓名	電話
陳思源	(02)2345-6789
吳文	(02)3456-6789
黃子文	(02)2234-3789

```
\usepackage{booktabs}
\begin{center}
\csp{3cm}{\ctxfk 同學會通訊錄}\[2pt]
\begin{tabular}{lc}
\toprule
\csp{1cm}{姓名} & \csp{1.5cm}{電話}\
\midrule
\csp{1cm}{陳思源} & (02)2345-6789 \\
\csp{1cm}{吳文} & (02)3456-6789 \\
\csp{1cm}{黃子文} & (02)2234-3789 \\
\bottomrule
\end{tabular}
\end{center}
```

圖 10.9: 調整中文字距

同學會通訊錄

姓名	電話
陳思源	(02)2345-6789
吳文	(02)3456-6789
黃子文	(02)2234-3789

圖 10.10: 加大中文標題之字距

圖 10.9 的例子中，標題長度定為 3cm，並以楷體字排版。我們將標題與表格全部置於 `center` 指令環境內，排版之後標題居於表格中央。除了標題之外，本例中之人名也是以 `\csp` 指令排版，寬度為 1 公分。如果姓名只有兩個字，排版時，兩字間距會自動拉開。

依同一原理，我們也可以設定固定之字距，譬如，下列巨集指令將字距拉大為 2mm，並給與一點的伸縮彈性：

```
\newcommand{\csp}[1]{\let\zori=z
\def\zx{\hskip 2mm plus0.2pt minus0.1pt}
\let\z=\zx \mbox{#1}\let\z=\zori}
```

利用此一指令，圖 10.9 排版標題之指令若改為：

```
\csp{\ctxfk 同學會通訊錄}\[2pt]
```

排版結果將如圖 10.10 所示。

以上指令的另一個應用是重新設定文稿之中文字距。`cwTeX` 內定之中文字距為 0pt，再加上一點伸縮彈性，但有人認為此一字距太小。欲加大中文字距，最簡單的方法是在全文設定區加上下列兩行指令：

```
\def\zx{\hskip 2pt plus0.2pt minus0.1pt}
\let\z=\zx
```

第 1 行定義一新指令 `\zx`，設定標準字距為 2pt；第 2 行指令使用 `\let` 指令將內定之字距改為新設定之字距。

以上方法雖然簡單，但如果每一篇文稿都希望加大字距，更簡單的方法是在執行 `cwtex` 時加上 `-z` 選項。譬如，執行 `cwtex` 時加入下列選項，其效果與上面兩行指令相同：

項目	分數	評述意見
方法	85	本研究所使用的方法是由作者本人所發展出來的。因此，作者對此一方法的運用頗為得心應手。
貢獻	88	從理論和實際應用來看，本研究可以說都很有貢獻。
文字	85	甚佳。

註：以上為虛構，如有雷同，純屬巧合。

```

\usepackage{booktabs}
\begin{tabular}{lcp{2.5cm}}
\toprule
項目 & 分數 & 評述意見 \\
\midrule
方法 & 85 & 本研究所使用 ...
...
... 頗為得心應手。\\[2pt]
貢獻 & 88 & 從理論和實際應用來看，
本研究可以說都很有貢獻。\\[2pt]
文字 & 85 & 甚佳。\\
\bottomrule
\end{tabular}
\par\smallskip
\parbox{.9\textwidth}{ 註：
以上為虛構，如有雷同，純屬巧合。}

```

圖 10.11: 表格內之文字段落

```
c:\xtemp>cwtext -z+0.2 test
```

如果你是使用 WinEdt 文字編輯軟體，直接在 [F9] 功能鍵之設定中加入上述選項，即可一勞永逸地重新設定字距。

10.2.5 表格內的文字段落

以上所舉例子，欄寬是由 \LaTeX 所決定。有時候，表格某一欄的內容是整段文字， \LaTeX 無法計算其寬度，欄寬必須自行設定。若想把某一欄位的寬度設為 2.5 公分，可以在 `tabular` 指令環境中選用 `p{2.5cm}` 選項，請見圖 10.11 之例子。

圖 10.11 的例子表格共有 3 欄，前兩欄由 \LaTeX 依各欄內之數字或文字決定其寬度，第 3 欄以 `p{2.5cm}` 指令固定為 2.5 公分。在固定寬度的表格欄裡，我們可以輸入多個段落。本例中第一欄位內僅包含一個段落。如果有兩個段落，第一段結束處應加上 `\par` 指令。要注意的是，第 2 段落行首並不會內縮；若有必要，我們可以在段落開始之處下 `\hspace*{...}` 指令，以產生適當的空白。

圖 10.11 同時說明如何以 `\parbox{...}` 指令在表格下面輸入註解說明。 \LaTeX 將 `tabular` 指令環境內全部文字視為一個大字母，因此排版註解之前，我們必須先以空白行或 `\par` 指令結束表格本身之段落。否則，註解將會排版在表格的右

方而不是底下。本例中，段落指令之後先以 `\smallskip` 指令拉大空白；其下之註解是以 `\parbox` 指令編排。因為是新的段落，因此起頭會自動內縮一點距離 (`indent`)。欲取消開頭之空白，應加上 `\noindent` 指令。第 10.6.3 節將介紹排版表格註解之 `threeparttable` 巨集套件。

圖 10.11 例子所使用的 `p{2.5cm}` 指令事實上等於是 `\parbox[t]{2.5cm}`，選項 `[t]` 設定將段落文字之上端對齊該行其他欄位之文字或數字。因此，第一行第 2 欄之數字“85”即和第 3 欄文字段落的首行對齊。我們也可以使用 `m{2.5cm}` 或者 `b{2.5cm}`。前者將段落文字的中央對齊其他各欄之文字或數字；後者則將段落文字之底端對齊其他各欄文字。為了對照比較，底下的表格例子中之段落是使用 `m{4cm}` 排版：

```
\begin{tabular}{lcm{4cm}}
```

我們另以 `\midrule` 指令加上橫線以區隔三項資料。請注意，`m{...}` 與 `b{...}` 是 `array` 巨集套件所提供的指令。欲使用這兩個指令排版，我們必須先在全文設定區引用巨集套件。

項目	分數	評述意見
方法	85	本研究所使用的方法是由作者本人所發展出來的。
貢獻	88	從理論和實際應用來看，本研究可以說都很有貢獻。
文字	85	甚佳。

10.2.6 其他控制與設定指令

表格排版還可使用畫線指令：`\hline`，`\vline`，與 `\cline`，其中，`\hline` 指令用於在表格內畫橫線。`\cline` 指令與 `booktabs` 巨集指令之 `\cmidrule` 類似，但功能較簡單。例如，`\cline{2-3}` 可用以排版橫跨 2–3 欄之短橫線。最後，`\vline` 指令用於在特定欄位處排版短垂直線，高度與欄高相同。若要排版橫跨兩欄位以上之文字，可使用多欄位指令：`\multicolumn`。`tabular` 指令環境另外還有一些排版與設定指令，合併列舉於表 10.1。請注意，部分指令是由 `array` 巨集套件所提供，欲使用這些指令須先引入巨集套件。

表 10.1: tabular 指令環境之指令 (array 巨集套件)

排版指令	
l	將文字資料排版於欄位左方,
c	將文字資料排版於欄位中央,
r	將文字資料排版於欄位右方,
	加上垂直直線,
p{...}	排版文字段落, 其頂端對齊其他各欄,
m{...}	排版文字段落, 其中央對齊其他各欄,
b{...}	排版文字段落, 其底端對齊其他各欄,
@{...}	將文字或宣告指令加於兩欄位中間, 並去掉原有之欄位間距,
>{...}	此指令可以加於 l,r,c,p,m,b 選項之前, 其功能是将括號內之宣告指令施加於欄位內文字之前,
<{...}	功能類似上記指令, 但將括號內之宣告指令施加於欄位內文字之後,
!{...}	此指令擴充上記 之功能。若以此項指令替代 , 排版之後出現的不是直線, 而是括號中之文字或符號。
設定指令	
\tabcolsep	tabular 指令環境下兩欄位間距之一半值, 內設為 6pt。
\arraycolsep	array 指令環境下兩欄位間距之一半值, 內設為 5pt。
\arrayrulewidth	直線與橫線條之粗細, 內設為 0.4pt。
\doublerulesep	連線兩直線 () 或兩橫線 (\hline\hline) 之間距, 內設值為 2pt。
\extrarowheight	此設定值用以加大橫行之間距, 但其深度 (depth) 維持不變。換言之, 僅基線上方之距離加大。
\arraystretch	設定行距之倍數。譬如, 若設定為 1.5, 則橫行間距將為原有之 1.5 倍。內設值為 1。

□ Brazil	33.92%
□ China	8.82%
□ India	8.37%

```

\usepackage{array,latexsym}
\begin{tabular}{!{\Box}%
>{\sffamily}lr<{\%}}
\toprule
Brazil & 33.92 \\
China & 8.82 \\
India & 8.37 \\
\bottomrule
\end{tabular}

```

圖 10.12: 表格指令之應用

以下以幾個例子說明表格指令之用法。首先，我們介紹 `!{...}`，`>{...}` 與 `<{...}` 指令之應用。以上三個指令是 `array` 巨集套件所提供，故須於全文設定區引用之。`!{...}` 指令是垂直線 `|` 指令的擴充；在大括號中填入的任何文字或指令將替代原垂直線。圖 10.12 之表格例子中，輸入指令為 `!{\Box}`，因此三個小正方形出現於表格左邊原垂直線之位置。

相對而言，`>{...}` 指令的作用不是替代特定欄位原有之內容，而是附加於欄位內容之前；`<{...}` 指令則是將大括號內之指令或文字附加於欄位內容之後。本例中，附加於前之指令為 `>{\sffamily}l`，因此表格第一欄位居左，且以 `sans serif` 字體排版。附加於後之指令為 `r<{\%}`，因此本例中第 2 欄位內之數字靠右，其後自動附加 `%` 符號。

在 `tabular` 指令環境中，利用 `|` 設定指令可畫出垂直線，但其粗細不能自己控制。利用上圖介紹的 `!{...}` 指令可以畫出任意粗細之垂直線。譬如，上例中之指令若改為：

```
!{\vrule width 2pt}
```

則表格左邊三個小正方形將改變為粗細為 `2pt` 之垂直線。指令中之 `\vrule` 為 `TeX` 指令，`2pt` 設定粗細。

垂直線指令 `|` 或 `!{\vrule width 2pt}` 所畫出之直線由上至下貫穿整個表格。如果只要在某一橫列中間畫短直線，可使用 `\vline` 指令。圖 10.13 例子中使用兩個 `\vline` 指令，畫出貫穿兩橫列之垂直線。

表 10.1 指令中，除了 `\arraystretch` 指令必須以 `\renewcommand` 指令重新定義之外，其餘各指令可以在 `tabular` 指令環境之前直接設定新值。例如，若欄高要加大 `2pt`，兩欄位之間距要加大 `2pt`，指令如下：

單位	國立台灣大學	
主持人	陳大川	322-4433
聯絡人	張小河	733-3344
地址	台北市羅斯福路	

```

\extrarowheight=2pt
\begin{tabular}{|l|l|}
\hline
單位 & 國立台灣大學 \\
\hline
主持人 & 陳大川 \\
\hline
聯絡人 & 張小河 \\
\hline
地址 & 台北市羅斯福路 \\
\hline
\end{tabular}

```

圖 10.13: 表格中短直線

```

\extrarowheight=2pt
\addtolength{\tabcolsep}{1pt}

```

如果要把橫距變為原來的 1.2 倍, 請在全文設定區加入下列指令:

```

\renewcommand{\arraystretch}{1.2}

```

10.3 tabularx 巨集套件

對應於 `tabular` 指令環境, 另有一個 `tabular*` 指令環境, 兩者不同的地方在於後者可以設定表格的寬度。若表格有 3 欄, 而排版指令為:

```

\begin{tabular*}{6cm}{lrr}

```

則表格寬度將為 6 公分, 每一欄之寬度仍由 \TeX 各欄之文字或數字自動計算大小。但是, 因為表格寬度已設定為 6 公分, 因此 \TeX 自動把欄距拉大, 以使表格寬度等於設定之值。顯然, 如果寬度設定值不適當, 排版結果不會太理想。

要排版固定寬度之表格, 另外一個辦法是使用 `tabularx` 巨集套件, 作者為 David Carlisle。此巨集套件亦讓排版者指定表格寬度, 但它的特點是可以自動算出特定欄位之寬度。換言之, 在此指令環境之下, 表格欄距仍維持內設值, 各欄之寬度可以自動調整。反之, `tabular*` 指令環境則是固定欄位寬度, 欄位間距則自動調整以達到選定的表格寬度。

圖 10.14 之表格例子是使用 `tabularx` 巨集指令所排版。此例中, 另外使用了

製造商	今年 排名	去年 排名
中國鋼鐵	1	2
宏 電腦	2	3
南亞塑膠	3	1
建元電子	4	4

```

\usepackage{tabularx,array}
\begin{tabularx}{4.5cm}{%
!{\vrule width 2pt}%
X|cc!{\vrule width 2pt}}
\hline
製造商 & 今年 & 去年\\[-2pt]
& 排名 & 排名\\
\hline
中國鋼鐵 & 1 & 2 \\
宏 電腦 & 2 & 3 \\
南亞塑膠 & 3 & 1 \\
建元電子 & 4 & 4 \\
\hline
\end{tabularx}

```

圖 10.14: tabularx 巨集套件

array 巨集套件之 `!{...}` 指令, 故須同時引用之。第一行指令甚長, 故拆為 3 行排出。首先, 表格寬度定為 4.5 公分, 內含三欄資料; 2、3 欄設定居中編排, 由 \TeX 自動計算欄寬。第一欄設定項為 `X`, 指示 \TeX 由表格寬度 4.5 公分減去 2、3 欄寬度之後, 其餘用以排版第一欄。本例中, 表格左右各畫了一條 2pt 粗細之垂直線, 使用之指令為 `!{\vrule width 2pt}`。此一指令之功能上一小節已說明, 此處不再重覆。

使用 `tabularx` 巨集套件時, 我們也可以設定將每一欄都以 `X` 選項編排。譬如, 若以下列指令排版表格:

```
\begin{tabularx}{6cm}{|X|X|X|}
```

則 \TeX 會將 6 公分平均分到 3 個欄位去。因為設定各欄位之前後畫有垂直線, 而直線本身有寬度, 因此每一欄位之寬度將比 2 公分略小一些。當表格之欄位設定以 `X` 指令排版時, 欄位內之文字將以上一節介紹之 `p{...}` 指令, 也就是 `\parbox[t]{...}` 指令編排。如果希望欄位內之文字以 `m{...}` 指令格式編排, 必須在 `tabularx` 指令環境之前以下列指令重新定義:

```
\renewcommand{\tabularxcolumn}[1]{>m{#1}}
```

重新定義之後, `X` 指令欄位下之文字段落將以 `m{...}` 方式排版。亦即, 欄位內之文字段落中間會對齊其他欄之文字或數字。

10.4 表格標題與位置

專業排版中，較大的圖表通常是移於在版面的上下端，或者單獨占一整頁。L^AT_EX 提供 `figure` 與 `table` 指令環境；前者用於控制圖位置，後者用以控制表格之位置。在 L^AT_EX 中，這兩個指令環境合稱為 `float` (浮動版面)；排版時，指令環境內之文字圖表將獨立安排於版面頁之上下方，或自成一頁。

除了控制圖表位置外，指令環境內尚可使用 `\caption` 指令編排標題。在控制圖表位置的功能上，這兩個指令環境並無不同，唯一的差別是在排版標題時，`table` 指令環境內之標題會自動加上 `Table` 一字；`figure` 指令環境下則自動加上 `Figure` 英文字。本章主要介紹表格之排版，但因為 `figure` 與 `table` 指令環境大同小異，故在此一併介紹。

10.4.1 浮動版面指令環境

浮動版面內通常置放排版圖或表格之指令，但也可以是單純的文字段落。使用 `table` 指令環境控制表格位置，指令如下：

```
\begin{table}[ pos]
...
\caption{...}
\end{table}
```

所有排版表格之指令即置於指令環境中。本例中，標題指令 `\caption` 置於指令環境末端，實際上它可以置於指令環境之前端，甚或任何位置。本例中，`table` 指令環境內用於排版表格。若是置放圖形，`table` 應改為 `figure`。如果文稿是以兩欄式 `twocolumn` 編排，則浮動版面須使用 * 型式，亦即 `table*` 與 `figure*`。

顧名思義，浮動版面是將特定文字段落或圖表另找位置排版。L^AT_EX 決定圖表的位置時，首先計算圖表本身的高度，其次要了解本頁版面剩下多少空間等等。即使圖表可以擠入版面中，但若剩餘供排版文字的空間太小，如此安排也不適當。L^AT_EX 是透過一套複雜的計算過程，才決定浮動版面的位置。欲了解詳情，請見 [Reckdahl \(1997\)](#) 或 [Goossens et al. \(1994\)](#)，第6章。在計算圖表之位置時，L^AT_EX 會參考一些控制值。有些人認為依 L^AT_EX 內定值所排版出來的圖表位置使版面太顯得空盪，如果你也有同樣的感覺，可試改用以下之設定值：

```
\renewcommand{\textfraction}{0.15}
\renewcommand{\topfraction}{0.85}
\renewcommand{\bottomfraction}{0.65}
\renewcommand{\floatpagefraction}{0.60}
```

這些修正值是許多使用者長時間測試之後推薦使用的，設定值須於全文設定區定義。

如果對於 \LaTeX 的選擇不滿意，可在 `table` 與 `figure` 指令環境中的位置選項 `pos` 另行設定。位置選項變數如下：

- `h` (here): 圖表置於現址，
- `t` (top): 圖表置於本頁上端，
- `b` (bottom): 圖表置於本頁下端，
- `p` (page): 圖表自成一頁。

以上 4 個位置選項變數可以進一步組合。如果不加選項，內定值為 `[tbp]`。簡單來說，這指示 \LaTeX 盡量設法將表格置於版面上端；若不成，則置於版面下端；若再不成，則讓表格自成一頁。

下指令時，位置選項變數之順序無關緊要， \LaTeX 永遠依照 `h-t-b-p` 之順序尋找適當位置。例如，不管選項是 `[bh]` 或 `[hb]`，尋找位置之順序都是 `h-b`。如果選項中僅有一個變數，如 `[t]`，`[b]`，或者 `[h]`，排版時可能出現問題；因為可供選擇的彈性太小， \LaTeX 有可能將圖表移至文稿最後一頁。因此，選項越多， \LaTeX 越能妥善處理。一般而言，不加任何選項已能獲得良好結果。如果要 \LaTeX 首先考慮將圖表置於行文中，可試用 `[htbp]` 或 `[htp]` 等。有關於浮動圖表指令環境之使用，請參見 [Reckdahl \(1997\)](#) 之說明。

如果你設計的版面特別，可在下位置選項變數時加入 `!`，例如：

```
\begin{table}[!ht]
```

此項設定要求將表格盡可能地置於行文指令處，若不成則置於版面上方。

10.4.2 圖表標題

浮動版面指令環境之內可以使用 `\caption{...}` 排版標題。在 `table` 指令環境中，標題之前會自動加上 `Table` 英文字並編入號碼，如 “Table 3:”；其後才是表格

標題。若是 `figure` 指令環境，標題前之字樣為 “Figure 9:”。

標題指令可置於圖表上方或下方，標題與圖表之間距有內設值。L^AT_EX 假設標題是置於圖表下方，排版時標題上方會自動拉大10pt 間距，下方則無額外空白。如果要把標題排版於表格上方，則標題之下應留一些空白，上方空白反而是不必要的。若內設之間距不適用，也可重新設定。若標題是置於圖表下方，在全文設定區加入下列指令即可重新設定間距為15pt:

```
\abovecaptionskip=15pt
\belowcaptionskip=0pt
```

如果圖與表採不同方式排版標題，圖標題置於下方，表標題置於上方，則以上之設定並不適用。解決方法是定義一 `\topcaption` 指令排版表標題:

```
\newcommand{\topcaption}{%
\setlength{\abovecaptionskip}{0pt}%
\setlength{\belowcaptionskip}{10pt}%
\caption}
```

定義指令後，圖標題仍以 `\caption` 指令編排，表標題則改用 `\topcaption` 指令排版。

如上所述，`table` 與 `figure` 指令環境唯一的差別是在於標題指令所加上之字樣不同。若排版英文稿，L^AT_EX 自動加入之 `Table` 與 `Figure` 字樣並無問題。但在中文稿中，較理想之表格標題是以「表 3:」起頭；圖標題以「圖 9:」起頭。修改辦法很簡單，只須在全文設定區加入以下兩道指令即可:

```
\renewcommand{\tablename}{ 表}
\renewcommand{\figurename}{ 圖}
```

以上指令中，`\tablename` 原定義為 `Table` 一字，現由指令改為中文字「表」。如果要選用圓體字，則指令最後應改為 `{\ctxfr 表}`。

使用 `\caption` 指令排版標題時，圖表會自動編上號碼。在 `book` 文件類別下，若本章為第 10 章，圖表之號碼將以 10.1, 10.2 之形式出現。若要改變圖表編號之字體與格式，最簡單的方法是使用 `caption2` 巨集套件，作者為 Harald Sommerfeldt。欲使用 `caption2` 巨集套件，須在全文設定區加入下列指令:

```
\usepackage[ option]{caption2}
```

大部分的設定是經由加入選項為之。

圖 10.15 列出 `caption2` 巨集套件之選項，其下為輸入之指令。選項可分為四類，第一類是控制標題排版方式。若標題文字超過一行，加入第一類選項即可控制排版方式。譬如，`normal` 格式是一般排版文章段落的方式，亦即，右邊將切齊；而且第一行並不內縮。`center` 選項之格式是標題各行皆居中，`centerlast` 則是最後一行居中，前面各行以 `normal` 方式編排。如果圖表標題不超過一行長度，標題將居中編排，不依以上之設定格式。若單行之標題也要與多行標題同樣方式編排，選項中須加入 `nooneline`。

第 2、3 類之指令是控制編號字體及大小。所謂「編號字體」是指標題文字前諸如 “Table 10.3:” 字串。字體選項中若選用 `sf`，則上述字串將排版為 “Table 10.3:”。若再加入 `large` 選項，字體將加大一些。若標題中文字是以粗黑體或圓體排版，則加入 `sf` 選項可使標題編號數字之字體與標題中文字較為一致。另外，若 “Table 10.3:” 要改為 “Table 10.3.”，可在全文設定區加入：

```
\renewcommand{\captionlabeldelim}{.}
```

本例子之標題選用 `normal` 選項，因此標題文字左右兩邊將切齊。不過，本例之標題文字雖然占 2 行，事實上僅有第一行是真正的標題，第 2 行是表格說明。為了排版說明文字，指令中使用了換行指令。`\caption` 指令內雖然可使用換行指令，不過，某些 \LaTeX 指令不能直接用於浮動版面指令環境內，否則會出現錯誤，換行指令就屬於這一類。若非要使用不可，其前須加上 `\protect`。故本例之換行指令變成 `\protect\\[2pt]`。

圖 10.15 例子中，第 4 行指令為 `\setcaptionwidth`，其作用是設定標題文字所占之最大長度，本例子設定標題寬度為行長的 0.77。`caption2` 巨集套件之其他設定指令，請參考 [Reckdahl \(1997\)](#)。

10.5 引述表格

圖 10.15 中，`\caption` 指令之後緊接著一行 `\label{tabcap}` 指令，其中 `tabcap` 是使用者自行定義的標誌 (marker)，目的是方便引用圖表。請注意，`\label` 指令須緊接在 `\caption` 指令之後。

標題排版格式	normal, center, flushleft, flushright, centerlast, hang, indent
編號字體大小	scriptsize, footnotesize, small, normalsize, large, Large
選用編號字體	up, it, sl, sc, md, bf, rm, sf, tt
單行標題	online, noonline

圖 10.15: caption2 巨集套件

此巨集套件可控制標題排版格式、字體與編號。

```

\usepackage[noonline,normal]{caption2}
\begin{table}
\centering
\setcaptionwidth{.77\textwidth}
\begin{tabular}{lm{.63\textwidth}}
\toprule
標題排版格式 & ...
...
單行標題 & |online|, |noonline|\\
\bottomrule
\end{tabular}
\caption{\textsf{caption2} 巨集套件\protect\\[2pt]
{\small
此巨集套件可控制標題排版格式、字體與編號。}}
\label{tabcap}
\end{table}

```

定義了標誌之後，文稿內任何地方都可使用 `\ref{tabcap}` 指令引用圖表；或以 `\pageref{tabcap}` 指令引用頁碼。舉例言之，行文中欲排版下列句子：

...如第 208 頁之圖 10.15 所示 ...

輸入指令如下：

如第\Z\pageref{tabcap}\Z 頁之圖\Z\ref{tabcap}\Z 所示 ...

TeX 會自動計算表編號與頁碼，代入句子中。請注意，排版時須執行 latex 兩次，才能計算出正確的頁碼與圖表編號。

以上所輸入的文字中，標誌指令前後所加上之 \Z 是 cwTeX 控制間距之指令，

目的是調整中文字與阿拉伯數字之間距。若不加上 `\z`，仍可得到結果，但間距會嫌太小。

10.6 表格排版細節調整

以上所介紹之指令與巨集套件已能處理絕大部分之表格排版。不過，有一些細節還是不容易處理。譬如，表格中之數字中若有小數點或逗號，上下列數字之小數點或逗號應對齊。圖 10.2 (頁 188) 曾說明如何使用 `~` 指令作人為調整；表 10.6 (頁 194) 則說明以 `@{.}` 指令排版上下對齊之小數點。但是，前一方法並不精確；後一方法可得到精確結果，但指令稍嫌麻煩。

若文稿中的表格有許多小數點數字，欲讓小數點上下對齊，可使用 `dcolumn` 巨集套件。另外，有些表格第 2 欄有兩項資料，而第 1 欄僅有一項資料。排版時，第 1 欄資料須水平對準第 2 欄上下兩列資料中間。欲排版此種表格，可使用 `multitrow` 巨集套件。如果表格下方要加上說明、註解，或者引述資料來源，最簡單的方法可能是使用 `threeparttable` 巨集套件。本節介紹一些巨集套件以解決這些林林總總的問題。

10.6.1 表格小數點對齊

表格中的數字若有小數點或逗點等，為求美觀起見，這些符號位置應上下對齊。表 10.6 (頁 194) 曾說明如何使用 `@{.}` 設定指令來排版上下對齊的小數點，但是其方法稍嫌複雜。本小節所介紹的 `dcolumn` 巨集套件提供同樣精確，但比較容易控制的方法。

在 `dcolumn` 指令環境內，可使用下列指令設定欲上下對齊之小數點：

$$D[\text{鍵入符號}][\text{輸出符號}][\text{小數點位數}]$$

所謂「鍵入符號」是指使用者在文稿內所鍵入之符號，「輸出符號」則指 \LaTeX 所排版出的符號。例如，我們可設定成鍵入逗號，但排版出小數點。指令中之「小數點位數」若設為 3，則排版時欄位之寬度應足以容納整數再加上 3 位小數。

為了方便起見，我們可以進一步使用 `\newcolumnmtype` 指令自行定義較簡化的設定指令。例如，如果鍵入與輸出符號都是小數點符號，我們可以定義如下之設定指令：

國民所得支用面		
項目	金額	比率
民間消費	\$13,665	46.7%
國內投資	5,066	17.32
政府消費	4,229	14.5

```

\usepackage{dcolumn,booktabs}
\newcolumntype{d}[1]{D{,}{,}{#1}}
\newcolumntype{.}[1]{D{.}{.}{#1}}
\begin{tabular}{cd{3}.{3}}
\multicolumn{3}{c}{國民所得支用面}\
\toprule
項目 &\multicolumn{1}{c}{金額}&
\multicolumn{1}{c}{比率}\
\midrule
民間消費 & \$13,665 & 46.7\rlap{\%}\
國內投資 & 5,066 & 17.32\
政府消費 & 4,229 & 14.5\
\bottomrule
\end{tabular}

```

圖 10.16: dcolumn 巨集套件

```

\newcolumntype{.}[1]{D{.}{.}{#1}}

```

以上指令所定義之設定指令，鍵入與輸出符號皆為英文句點，使用指令時小數點位數須填入指令選項中。

圖 10.16 提供一個應用例子。在此例子中，我們定義兩個設定指令 `.{...}` 與 `d{...}`，後者之鍵入與輸出符號都是逗號。使用 `d{...}` 與 `.{...}` 指令時，括號內應填入小數點位數。如果表格中上下各列的小數點位數不相同，原則上應填入最大的位數。舉例言之，圖 10.16 第 3 欄有上下 3 列數字，第 2 列數字 17.32 小數點有兩位，其他兩個數字只有一位。本例中，此欄是以 “`.{3}`” 排版，故上下之數字對齊小數點，而且欄位之寬度除了容納整數之外，還可容納 3 位之小數。

如果此欄改以 “`.{1}`” 排版，則 3 個數字還是會完整排出，但是欄位之寬度僅容納整數及 1 位小數。事實上，小數點位數之選項也可指定整數位數。例如，若此欄改以 “`.{3.2}`” 排版，代表整數部分有 3 位，小數部分有 2 位；排版時整數部分空出可容納 3 位數之空間，小數部分空出可容納 2 位數之空間。反之，如果選項部分填入 “`.{-1}`”，則 `TeX` 將空出所有小數中最大位數之空間，而且，整數部分也取同樣空間，故小數點將排於該欄位的正中央。

圖 10.16 的表格是以底下指令排版：

```

\begin{tabular}{cd{3}.{3}}

```

表格計有 3 欄；第 1 欄之文字居中排版，2、3 欄內都是數字。其中，第 2 欄三個數字

年期	產量
1991—1992	934,345
1992—1993	1,134,045

```

\usepackage{dcolumn}
\newcolumntype{a}[1]{D{-}{-}{#1}}
\newcolumntype{b}[1]{D{.}{,}{#1}}
\begin{tabular}{|a{4}|b{3}|}
\hline
\multicolumn{1}{|c|}{年期} &
\multicolumn{1}{c|}{產量} \\ \hline
1991-1992 & 934.345 \\
1992-1993 & 1,134.045 \\ \hline
\end{tabular}

```

圖 10.17: 表格數字上下對齊

都是整數，中間夾有逗號。為了使逗號上下對齊，第 2 欄以 `d{3}` 來設定，其中的 3 指示「小數點」之後有 3 位數。從這個例子可以看出來，所謂的「小數點」，其實可以是任意的指定符號。第 3 欄的數字中整數有兩位，小數有一位，但是第一個數字之後有百分比符號。本例以 `.{3}` 設定小數點應上下對齊，讓小數部分留出一個空白。因為百分比符號並非數字，如果直接鍵入，排版時數字將無法對齊小數點。本例中，我們使用 \TeX 的指令 `\rlap` 來排版百分比符號。其功能是在 46.7 之後排出百分比符號，然後再退回到數字 7 之右緣。如此一來， \TeX 在對齊小數點時，不會考慮到百分比符號的存在。與 `\rlap` 對應的指令是 `\llap`；前者處理右邊之文字符號，後者處理左邊的文字符號。

有人可能會覺得奇怪，為何要區別鍵入與輸出符號？圖 10.17 的例子說明此種區別之用途。此表格中有兩欄數字，其中，第 2 欄上下兩個數字位數不同，上面為 934,345，底下為 1,134,045。我們定義兩個設定指令 `a{...}` 與 `b{...}`。在 `a{...}` 設定指令中，鍵入與輸出符號同樣都是一短線（連字號）。但是在 `b{...}` 設定指令中，鍵入符號定義為小數點，輸出符號則為逗號。

如果將鍵入與輸出符號都定義為逗號，排版時上面數字之逗號將對齊底下數字左邊的逗號，結果並不正確。為解決此一困難，我們利用 `b{...}` 設定指令，但鍵入之數字須改為 934.345 與 1,134.045。排版之後，小數點會改為逗號，並且上下對齊而得到正確的結果。

10.6.2 橫列文字對齊

前面所介紹的 `\multicolumn` 指令是用於排版橫跨兩欄以上的標題。有時候，表格中某欄位之數字可能占用數列空間。譬如，圖 10.18 例子中，左欄之「投資」對

投資	政府部門 民間企業
儲蓄	家庭部門

```

\usepackage{multirow,booktabs}
\begin{tabular}{ll}
\toprule
\multirow{2}{1cm}{\ctxfk 投資} &
    政府部門 \\
    & 民間企業 \\
\midrule
{\ctxfk 儲蓄} & 家庭部門 \\
\bottomrule
\end{tabular}

```

圖 10.18: multirow 巨集套件

應右欄之「政府部門」與「民間企業」。欲排版此種表格，可使用 `multirow` 巨集套件，這是 Terry Leichter 與 Piet van Oostrum 合作之成果。

此一巨集套件提供 `\multirow` 指令，內容如下：

```
\multirow{ nrows}{width}[fixup]{text}
```

第一選項 `nrows` 設定占用多少欄位。本例中，「投資」兩字占用兩橫列。第二選項設定欄位寬度，本例中為 1 公分。如果欄位寬度要由 \TeX 計算決定，則 `{1cm}` 應以 `*` 替代。第 3 選項 `[fixup]` 可有可無，其功能是調整垂直位置。本例中未加入此選項，故不調整。若改加上 `[2pt]`，則「投資」二字將上移 2pt；若填入負值，則文字將往下移。最後一個選項是排版文字。在第一選項之後還可加入一控制垂直間距之選項，詳情請見巨集套件內附之說明。

圖 10.19 是一個較複雜的例子，但指令之使用方法並無不同。本例子中有幾個地方都須設定橫列文字對齊，為了避免重複輸入指令之麻煩，我們在全文設定區定義一簡單巨集指令 `\mr`，其中設定排版文字將跨占兩橫列空間；欄位寬度由 \TeX 自動計算。定義 `\mr` 指令之後，排版指令即大幅簡化，較不易發生錯誤。

10.6.3 表格註解

圖 10.11 (頁 198) 曾說明如何以 `\parbox` 指令排版表格下方之說明或註解。此一方法須自行設定寬度，使用上並不方便。欲在表格下方加上註解，較方便的方法是使用 Donald Arseneau 所寫的 `threeparttable` 巨集套件。此巨集套件會自動計算表格寬度，減少來回調整的麻煩。

巨集套件 `threeparttable` 提供一同名的指令環境；此指令環境可視為是 `table`

		未然形	連用形	終止形	連體形	假定形
例	語幹	-	-	-	-	-
寒	寒					
暑	暑					

```

\usepackage{multirow,booktabs}
\newcommand{\mr}[1]{\multirow{2}*{#1}}
...
\begin{tabular}{cccccc}
\toprule
&& 未然形 & 連用形 & 終止形 & 連體形 & 假定形 \\
\mr{例} & \mr{語幹} & \mr{-} & - & & & \\
& \mr{-} & \mr{-} & \\
&& - & & & & \\
\midrule
寒 & 寒 & \mr{ } & & \mr{ } & \mr{ } & \mr{ } \\
暑 & 暑 & & & & & \\
\bottomrule
\end{tabular}

```

圖 10.19: 對齊橫列文字 — multirow 巨集套件

與 figure 浮動版面指令環境功能之延伸。浮動版面指令環境可將大型圖表排版於版面適當位置，指令環境內尚可使用 \caption 指令排版標題；threeparttable 指令環境亦有類似功能。圖 10.20 例子之內容與圖 10.16 (頁 210) 相同，但新增加了表格註解與說明。指令環境內以 \caption 指令排版標題；以 tabular 指令環境排版表格。

排版表格註解時，先以 \tnote 指令標示符號，表格下方則以 tablenotes 指令環境排版註解。請注意，表格之註解不會自行編號，使用者須自行輸入編號或符號。此一例子中，事實上僅有一個註解，以 * 號表示。表格下方使用 \item[*] 指令排版註解。本例尚使用 \item[] 指令排版一段說明文字。 \item 指令後緊接兩個左右中括號表示不加入任何標示符號。不過，如此一來， \item 指令後面之文字仍會往右移動若干距離。為了使說明文字段落的第一行往左突出一點，我們在第一個 \item[] 指令之後加上 \hspace*{-10pt} 指令，使行文字往左移 10pt。

項目	金額	比率
民間消費	\$13,665	46.7%
國內投資*	5,066	17.3
政府消費	4,229	14.5

說明: 金額為新台幣億元。資料來源: 《中華民國國民所得統計》。
* 國內投資包括政府投資與民間投資。

```

\usepackage{threeparttable}
\begin{threeparttable}
\caption{ 國民所得支用面}
\begin{tabular}[c]{cd{3}.{3}}
...
民間消費 & \$13,665 & ...
國內投資\tnote{*} & 5,066 & ...
政府消費 & 4,229 & 14.5
\bottomrule
\end{tabular}
\begin{tablenotes}
\item[] \hspace*{-10pt} 說明:
金額為新台幣億元。...
\item[*] 國內投資 ...
\end{tablenotes}
\end{threeparttable}

```

圖 10.20: 使用 threeparttable 排版表格註解

10.6.4 表格內加入括弧或斜線

有些表格須加入大括弧或斜線，本節所介紹之巨集指令 `\bpara` 可以方便地處理此種情況。此一巨集指令取自 [Goossens et al. \(1997, 頁 49\)](#)，指令之說明請見原書。簡單言之，我們利用了 `graphicx` 巨集套件所提供之旋轉功能，將選定之符號加於表格內。巨集指令 `\bpara` 必須定義於全文設定區內，同時我們還須引用 `graphicx` 巨集套件。

巨集指令之內容如下：

```

\usepackage{graphicx}
\newcommand{\bpara}[4]{ % #1 x; #2 y; #3 angle; #4 height
\begin{picture}(0,0)%
\setlength{\unitlength}{1pt}%
\put(#1,#2){\rotatebox{#3}{\raisebox{0mm}[0mm][0mm]{%
\makebox[0mm]{\left.\rule{0mm}{#4pt}\right\}}}}%
\end{picture}}

```

`\bpara` 指令計有 4 個選項，分別設定大括號之置放位置、大小與旋轉角度。各選項之意義分別說明如下：

1. 橫向移動距離,
2. 縱向移動距離,
3. 旋轉角度,
4. 括號大小。

移動距離與括號大小之單位皆為 pt; 移動距離之值可正可負, 若填入負值, 表示反方向移動。

圖 10.21 的例子中, 我們在表格中加入 4 個右大括號, 其長度視涵蓋內容而有所不同。為了節省空間, 例子內不再重覆巨集指令之定義。第一欄有三個大括號, 第一個大括號置於「基隆廳」三個字右邊, 上移 2pt, 長度為 17pt。中間大括號長度為 17pt, 但該欄位內並無任何文字, 經過幾次嘗試, 我們發現若右移 34pt, 大括號可以上下對齊, 故指令為 `\bpara{34}{0}{0}{17}`。其餘兩個大括號之指令也是反覆調整後決定的。

此一表格共有三欄, 皆靠左編排。如果我們改為靠右編排, 則加入 `\bpara`

表 1: 行政區域劃分變遷		
1905	1915	1920
台北廳	} 台北廳	} 台北州
基隆廳		
深坑廳		
宜蘭廳	} 宜蘭廳	
桃園廳		
新竹廳		

```

\caption{ 行政區域劃分變遷}
\setlength{\extrarowheight}{2pt}
\begin{tabular}{l@{\hspace{4pt}}%
l@{\hspace{4pt}}l}
\toprule
~1905& ~1915& ~1920\\
\midrule
台北廳 & & \\
基隆廳 \bpara{0}{2}{0}{17} & & \\
& 台北廳 & \\
深坑廳 & & \\
\bpara{30}{0}{0}{17} & & 宜蘭廳 \\
\bpara{0}{0}{0}{35} & & 台北州\\
宜蘭廳 & & \\
桃園廳 \bpara{0}{0}{0}{17} & & \\
& 桃園廳 & \\
新竹廳 & & \\
\bottomrule
\end{tabular}

```

圖 10.21: 表格內之大括號

$x \backslash y$	y_1	y_2
1	34	55
2	25	45

```

\begin{tabular}{l@{r}rr}
& $y$ & & \ll[-9pt]
& & $y_1$ & $y_2$ \ll[-9pt]
\hline
1 & 34 & 55 \\
2 & 25 & 45
\end{tabular}

```

圖 10.22: 表格內斜線

指令後, 該欄位內容長度加大, 各欄位之文字無法上下對齊。解決的方法是使用 `\rlap` 指令, 請參見前面例子之說明。

除大括號之外, 類似的指令可以用來排版箭頭、方括號等等。前面 `\bpara` 指令之定義中, 倒數第 2 行末端有 `\}` 指令, 其功能就是設定排版大括號。若將此改為 `]`, 則同樣的指令將排版出方括號。同理, 若改以 `\rangle`, 結果為一右三角形括號。事實上, 我們可以使用表 9.13 (頁 177) 所介紹之任何一個界限符號 (delimiter)。譬如, 若想要排版箭號, 只要將 `\}` 改為 `\uparrow` 即可。定義更改之後, 使用下列指令:

```
\bpara{0}{0}{-45}{20}
```

將排版出一長度 20pt, 箭頭朝右上方之箭號。若第三個選項填為 45, 箭頭將變成朝右下方。

利用同樣的原理, 畫出任意角度與長度之斜線也很容易。將 `\bpara` 巨集指令之定義複製一份, 改名為 `\bline`, 並將倒數第 2 行改為:

```
\makebox[0mm]{\rule{0.4pt}{#4pt}}}
```

直線之粗細可自行選擇, 本例中定義直線粗細為 0.4pt。利用以上定義, 即可在表格中加入斜線。圖 10.22 利用 `\bline` 指令在表格內加入一直線, 長度為 28pt, 角度為 55。

此一例子的目的是說明表格內斜線之畫法, 實際上, 加入斜線後, 表格並不見得更清楚, 也不見得更美觀。這印証前面所講的, 花費力氣排版複雜版面之前, 應先想想是否有更簡單美觀的設計。

10.7 彩色表格

排版的目的是精簡地傳達大量資訊，因此表格上應避免花花綠綠的色彩。不過，表格內若有大量數字，則適當地加上灰階橫條的確有助於閱讀。另外，適當地方加上彩色也有醒目效果。欲在表格中加入灰階/色彩，可使用 David Carlisle 之 `colortbl` 巨集套件。

圖 10.23 說明如何在表格中加上灰階橫條紋，使用的指令是 `\rowcolor`。此一指令可將特定橫列加上彩色或灰階，指令格式如下：

```
\rowcolor[ color model]{color}[left overhang][right overhang]
```

所謂 *color model* 是 `color` 巨集套件選定色彩的方式，請見第 12 章之說明。欲將某一橫列加上灰階，`\rowcolor` 指令須加在該列最前方。本例中，我們使用灰階 `gray` 模式，0.85 則用以選定灰階之深淺度。灰階度越接近零，顏色越深；越接近 1，灰階越淺。第 3 選項 *left overhang* 設定彩色或灰階凸出最左一欄左邊之尺寸；相對的，*right overhang* 則設定凸出最右欄右邊之尺寸。若不加設定，巨集套件會自動設定讓色彩或灰階與表格左右邊緣切齊。

除了加上灰階橫條紋之外，本例中之細橫線也是灰階線條，指令為：

```
\arrayrulecolor{light}
```

其中，`light` 是事先以 `\definecolor` 指令所定義之灰階色，粗淺度為 0.85。請注意，`\arrayrulecolor` 指令後面之表格橫線與直線全部改變為灰階。如果表格線條要回復純黑色，首先須定義一純黑色，例如 `pureblack`，其灰階度等於 0，然後再以 `\arrayrulecolor` 指令回復黑色。

除了 `light` 之外，本例尚定義 `title` 為 0.30 之灰階，用於排版表格標題「通訊錄」三個字。任何文字要加上灰階或色彩，可使用 `\textcolor` 指令，詳細說明請見第 12 章。本例中，除了將文字加上灰階之外，同時也使用 `\csp` 指令將三個中文字之間隔拉大，標題長度訂為 2 公分。此一指令須在全文設定區定義，方法請見圖 10.9 之例子。

對應於 `\rowcolor` 指令，`\columncolor` 指令可將表格特定欄加上色彩/灰階，指令如下：

```
\columncolor[ color model]{color}[left overhang][right overhang]
```

姓名	電話	研究室
毛正之	529	E302
古芸安	526	E109
宋名涵	531	E304
李真如	528	E205

```

\usepackage{colortbl}
\definecolor{light}{gray}{.85}
\definecolor{title}{gray}{.30}
\arrayrulecolor{light}
\begin{center}
\textcolor{title}{\csp{2cm}%
{\ctxfbb 通訊錄}}
\begin{tabular}{lll}
\hline
姓名 & 電話 & 研究室 \\
\hline
毛正之 & 529 & E302 \\
\rowcolor[gray]{.85}
古芸安 & 526 & E109 \\
宋名涵 & 531 & E304 \\
\rowcolor[gray]{.85}
李真如 & 528 & E205 \\
\hline
\end{tabular}
\end{center}

```

圖 10.23: 彩色表格例1

各選項之意義與 `\rowcolor` 相同。此一指令之使用須利用 `array` 巨集套件所提供之 `>{...}` 指令之功能。圖 10.24 將表格第 1 欄加上黃色背景, 其方法是在 `tabular` 指令環境中設定各欄位排版方式時使用下列指令:

```

\begin{tabular}>{\columncolor{yellow}}cccc}

```

上述指令中, `yellow` 是 `color` 巨集套件內定之顏色, 而 `colortbl` 巨集套件選擇顏色的方法與 `color` 巨集套件相同。其他顏色之選定方法, 請見第 12 章之說明, 或 `color` 巨集套件之說明檔。本書是以黑白列印, 因此黃色之背景是以灰階出現。

圖 10.24 之標題行另外以 `\rowcolor` 指令加上灰階背景。由排版結果可以看出來, 當橫列與直欄同時加上色彩或灰階時, 橫列之色彩指令優先。本例表格中之小方塊是以 `Box` 指令排版, 前面加上 `\large` 指令, 稍微放大。因為小方塊重覆出現, 我們在全文設定區定義巨集指令 `\B`, 代表此一小方塊。利用巨集指令, 一方面減少重覆輸入的麻煩, 另一方面也避免輸入時出現錯誤。請注意, 使用 `Box` 指令時, 須引用 `latexsym` 巨集套件。

項目	滿意	普通	不佳
準時上課	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
準備充分	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
講解清楚	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```

\usepackage{colortbl,latexsym}
\newcommand{\B}{\large$Box$}
\begin{tabular}{>{
\columncolor{yellow}}cccc}
\rowcolor[gray]{.85}
\hline
項目 & 滿意 & 普通 & 不佳
\hline
準時上課 & \B & \B & \B \\
準備充分 & \B & \B & \B \\
講解清楚 & \B & \B & \B \\
\hline \end{tabular}

```

圖 10.24: 彩色表格例2

如果要將整個表格加上彩色背景,可使用 `\colorbox`,此一指令是 `color` 巨集套件所提供。圖 10.25 例子沿用上面所定義之 `light` 灰階色,使用 `\colorbox` 指令將表格背景加上色彩。加上灰階或彩色背景時,表格四周要留出多大空間可以自行設定。例如,若在表格指令之前加入下列一行指令:

```
\fboxsep=10pt
```

表格四周即留出 10pt 之空間。`\colorbox` 指令不僅可以用於將表格加上彩色/灰階背景,也可施用於圖形或一般文字;請見第 12 章之例子。

常見的表格設計方法是以較淺顏色做為表格背景,再以較深的顏色排版文字標題。表格標題欲以彩色/灰階排出,可使用 `\textcolor` 指令,這也是 `color` 巨集

項目	滿意	普通	不佳
準時	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
準備	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
講解	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```

\usepackage{colortbl,latexsym}
\fboxsep=10pt
\newcommand{\B}{\large$Box$}
\colorbox{light}{
\begin{tabular}{cccc}\ctxfk
項目 & 滿意 & 普通 & 不佳 \\
\hline \ctxfm
準時 & \B & \B & \B \\
準備 & \B & \B & \B \\
講解 & \B & \B & \B \\
\hline \end{tabular}}

```

圖 10.25: 圖表加上灰階背景

套件之指令; 請見第 12 章之說明。

10.8 超大型表格

超大型表格是指表格寬度大於版面寬度, 或者表格高度超過版面高度。如果超出之尺寸不大, 最簡單的解決辦法是以較小字體排版表格。若縮小字體猶不能解決問題, 可考慮將特別寬的表格轉 90 度橫排, 特別長的表格則拆為兩三頁之接續表格。以下分別介紹表格橫排與排版超長表格的方法。

10.8.1 旋轉大型表格

欲旋轉大型表格至少有兩種方法, 第一種方法是使用 `\rotatebox` 指令, 這是 `graphicx` 巨集套件所提供之指令。第二種是使用 `lscape` 巨集套件。前者較適合旋轉小表格, 而且可旋轉任何角度。後者是將某頁版面由直排轉為橫排, 很適合用於旋轉占滿一頁的表格。`lscape` 巨集套件將於下一小節 (頁 220) 介紹, 此處先介紹 `\rotatebox`, 指令格式如下:

$$\backslash\text{rotatebox}\{ \textit{angle}\}\{\textit{material}\}$$

第一選項 *angle* 為旋轉角度, 第二組大括號內為旋轉之文字或圖表。除了選定旋轉角度外, 我們尚可設定旋轉之基準點, 請參見 12.1 節 (頁 243) 之說明。

使用此道指令須注意一點, 全部之文字與指令須置於單一段落內。亦即, 欲旋轉之表格內不得有空行或 `\par` 指令在內。若表格內必須留出空行, 解決辦法是將全部表格指令置於 `minipage` 指令環境內。圖 10.26 是旋轉大型表格的一個例子, 旋轉之角度為 90 度。排版此一表格使用了 `graphicx`, `booktabs`, 與 `threeparttable` 巨集套件。因為空間不足, 例子中並未列出引用巨集套件之指令。

10.8.2 超長表格

另一種類型的超大表格是其高度超過版面高度。前面曾說明, 若以 `tabbing` 指令環境排版表格, 其長度可以超過版面一頁; 但以 `tabular` 指令環境所排版之表格則不行。欲使用類似 `tabbing` 指令環境之指令排版超長表格, 可使用 `longtable` 巨集套件, 作者為 David Carlisle。

淡水與基隆港出口貨物量

	1899	1900	1901	1902	1903	1904	1905	1906	1907	1908
淡水	11,752	10,937	10,325	12,904	11,633	9,050	8,320	6,429	2,058	755
基隆	n.d.	n.d.	n.d.	939	2,832	4,070	5,836	6,625	10,441	11,597

* 基隆台北間舊鐵道線（在來線）於 1899 年度開始營運，1900 年改良線路通車，內含 6 哩「在來線」；同年舊線撤去。資料來源：臺灣總督府鐵道部（1910），下卷，頁 187-8, 218。

```

\rotatebox{90}{
\begin{minipage}{1.2\textwidth}
\begin{threeparttable}
\centering
淡水與基隆港出口貨物量\\
\begin{tabular}{lrrrrrrrrrr} \toprule
& 1899 & 1900 & 1901 & 1902 & 1903 & \dots & \dots & \midrule
淡水 & 11,752 & 10,937 & 10,325 & 12,904 & \dots & \dots & \dots & \dots & \dots & \dots \\
基隆 & n.d. & n.d. & n.d. & 939 & \dots & \dots & \dots & \dots & \dots & \dots \\
\bottomrule \end{tabular}
\begin{tablenotes}
\item[*] 基隆台北間舊鐵道線（在來線）... 下卷，頁 187--8, 218 。
\end{tablenotes} \end{threeparttable}
\end{minipage}}

```

圖 10.26: 旋轉大型表格

巨集套件 `longtable` 提供 `longtable` 指令環境，其指令格式與 `tabular` 很類似。最簡單的情況如下，如果你原先是以 `tabular` 指令環境排版表格，卻發現表格高度超過一頁，此時僅須於全文設定區引用 `longtable` 巨集套件，並將 `tabular` 指令替換為 `longtable`，原來之表格即自動拆為兩部分，分別排版於兩頁中。須注意的是，若使用 `longtable` 指令環境，排版須執行 `latex` 兩三次，甚至更多次，才能得到正確結果。

表 10.2 為排版超長表格的一個例子。(為調整排版空間，長表格置於本章最後面。)在 `longtable` 指令環境內也可以使用 `\caption` 指令排版標題；表格之編號將接續原來 `table` 指令環境之編號。必要時，我們也可以使用 `caption2` 巨集套件排版標題。`TeX` 的 `table` 指令環境是將表格置於版面適當地方，`longtable` 巨集套件所排版之表格長度超過一頁，自然沒有浮動版面之「能力」。

文稿內使用 `longtable` 指令時，表格將從該處開始排版。如果希望從下一頁頂端開始排版，則指令環境之前須自行加上 `\newpage` 指令。此外，表格每一頁之行數是 `TeX` 自行計算。若希望某一頁之表格要特別短一些，可以在拆頁處加上 `\pagebreak` 指令。

由本例子可見，表格欄位的設定方式、行距之控制、畫橫線等，都和 `tabular` 指令環境相同。不過，`longtable` 特別提供指令以處理上下兩頁表格之接續問題。本例中，`\endhead` 指令的功能是用以排版欄位標題，如「出口物價指數」、「進口物價指數」等等。`\endhead` 指令以上到另一個控制指令 `\endfirsthead` 之間的所有文字將出現為表格各頁上方之欄位標題，其中包含有「承接上頁」四個字。不過，表格第一頁不須有「承接上頁」，因此，第一頁之欄位標題須特別以 `\endfirsthead` 設定，重點是去除「承接上頁」四個字。

同樣的，`\endfoot` 與 `\endlastfoot` 指令是在表格每一頁下方排版出上下頁表格接續之說明。本例中，`\endfoot` 指令用於在每一頁下方排版出「續接下頁」四個字；`\endlastfoot` 則用於排版表格末頁底下之文字。本例中，`\endlastfoot` 與 `\endfoot` 之間空白，因此表格末頁底下即無任何添加資訊。

在 `longtable` 指令環境內可以使用 `\footnote` 排版註解，不過註解文字會出現在該頁下方，而不是表格末端。如果要在表格最末端加上說明或註解，可利用 `\endlastfoot` 指令。本例中，若在 `\endfoot` 與 `\endlastfoot` 兩行之間加入以下指令：

```
\multicolumn{6}{0.8\textwidth}{ 說明: ... }
```

則說明文字將排版於表格最後一頁下端。此一方法的缺點是排版說明文字段落的寬度須自行選定。

如果表格又長又寬,我們還可以使用 `lscape` 巨集套件將 `longtable` 所排版之長表格旋轉 90 度。前面曾介紹如何使用 `\rotatebox` 指令旋轉文字圖表,該指令可將整頁版面內一小部分的文字圖表旋轉任意角度。相對而言,`lscape` 巨集套件是將整頁版面旋轉 90 度,但頁眉/頁足不動,作者亦為 David Carlisle。

巨集套件 `lscape` 提供 `landscape` 指令環境,使用方法很簡單,僅須將所欲轉置之文字圖表置於指令環境即可。例如,要將表 10.2 旋轉 90 度,僅須將原排版指令納入 `landscape` 指令環內即可:

```
\usepackage{longtable,lscape}
\begin{landscape}
\begin{longtable}{@{}lrrrrr@{}}
...
\end{longtable}
\end{landscape}
```

最後,本章 10.2.3 節 (頁 194) 曾介紹 `booktabs` 巨集套件,可用以控制表格橫線之細節。不幸的是,原始版本的 `booktabs` 之指令無法與 `longtable` 巨集套件一起使用。不過,原作者 Simon Fear 特別提供一小段修改程式,我們已加入原巨集套件內。修正的版本置於光碟 `\support` 檔案夾內,請以之替代 `c:\texmf\tex\latex\booktabs` 內之原始版本。利用修正版本,如果要在 `longtable` 表格內畫上橫線,指令為 `\LTtoprule`,畫底橫線之指令為 `\LTbottomrule`,畫中間細橫線之指令為 `\LTmidrule`。

```

\usepackage{longtable}
\fontsize{10}{12.5pt}\selectfont
\extrarowheight=1pt
\begin{longtable}{@{}lrrrrr@{}}
\caption{ 台灣長期物價指數\label{longtable}}\[-2pt]
\hline
      & & 台銀躉售 & & 出口 & & 進口\[-1pt]
年期 & PPI & 物價指數 & CPI & 物價指數 & & 物價指數\
\hline
\endfirsthead
\multicolumn{6}{l}{\ctxfk 承接上頁}\[2pt]
\hline
      & & 台銀躉售 & & 出口 & & 進口\[-1pt]
年期 & PPI & 物價指數 & CPI & 物價指數 & & 物價指數 \
\hline
\endhead
\hline
\multicolumn{6}{r}{\ctxfk 續接下頁}
\endfoot
\endlastfoot
1896 & -- & -- & -- & 60.31 & & 59.16\
...
1945 & 490.09 & 2392 & -- & -- & -- \
1946 & -- & 21344 & -- & -- & -- \
\hline
\end{longtable}

```

表 10.2: 台灣長期物價指數

年期	PPI	台銀躉售 物價指數	CPI	出口 物價指數	進口 物價指數
1896	—	—	—	60.31	59.16
1897	—	—	—	66.88	55.66
1898	—	—	—	76.87	57.84
1899	—	—	—	77.60	62.27
1900	—	—	—	77.47	74.60
1901	—	—	—	78.00	77.21
1902	69.12	—	—	90.92	73.67
1903	73.16	—	65.44	82.91	76.66

續接下頁

承接上頁

年期	PPI	台銀躉售 物價指數	CPI	出口 物價指數	進口 物價指數
1904	60.58	-	68.88	85.77	79.81
1905	71.58	-	70.82	83.72	85.30
1906	72.67	-	71.45	84.74	89.26
1907	89.70	-	73.57	94.34	95.76
1908	77.96	-	79.08	98.62	87.83
1909	77.48	-	83.50	101.98	90.28
1910	88.97	-	86.18	93.64	95.25
1911	95.91	-	95.42	99.77	99.11
1912	110.36	-	104.17	113.43	99.57
1913	112.89	-	101.80	114.43	102.57
1914	100.00	-	100.00	100.00	100.00
1915	98.51	-	92.11	93.62	108.25
1916	108.93	-	97.53	98.05	144.69
1917	128.88	-	117.63	121.56	186.08
1918	166.89	-	146.43	151.58	220.07
1919	237.38	223	179.50	196.52	232.13
1920	253.80	264	158.14	233.51	251.96
1921	152.09	207	140.42	172.11	599.30
1922	127.23	203	131.00	166.07	545.90
1923	145.34	198	126.37	154.21	541.73
1924	157.65	198	135.49	162.67	568.81
1925	183.69	202	144.15	161.50	587.40
1926	147.39	188	140.65	159.70	506.30
1927	136.47	178	131.34	151.32	480.98
1928	133.25	180	133.47	150.32	478.92
1929	129.67	165	133.80	147.70	466.63
1930	110.49	151	119.03	129.18	388.95
1931	85.17	134	105.52	109.49	557.60
1932	94.46	140	100.79	70.16	326.53
1933	110.95	152	103.56	75.57	376.90
1934	114.82	151	106.21	82.94	375.77
1935	120.80	156	115.79	86.96	405.45
1936	130.44	166	124.34	96.49	385.65
1937	135.98	187	131.49	97.84	432.91
1938	138.26	216	165.05	125.06	468.77
1939	157.92	240	-	-	-
1940	173.08	270	-	-	-
1941	189.23	262	-	-	-
1942	202.10	296	-	-	-

續接下頁

承接上頁

年期	PPI	台銀躉售 物價指數	CPI	出口 物價指數	進口 物價指數
1943	210.70	315	-	-	-
1944	258.66	354	-	-	-
1945	490.09	2392	-	-	-
1946	-	21344	-	-	-

11 引用外製圖形

圖形是表達作者思想的有效工具，一般文稿或學術論文中經常使用圖形。T_EX 的主要功能是文字排版，它雖然可以繪製簡單的線條圖形，但複雜圖形就無能為力。同樣的，E_T_X 雖然新增一些畫圖指令，但仍然難以處理複雜圖形。不過，仍然有一些巨集套件如 PiCT_EX, E_T_XCAD 等，直接使用 E_T_X 的指令繪製圖形。

雖然 T_EX 的主要功能是處理文字，但是 Knuth 當初設計時留下一個與其他繪圖系統溝通的管道。T_EX 有一個指令稱為 `\special`，透過此一指令，我們可以將其他軟體繪製之圖形引入版面中。T_EX 並不瞭解其他繪圖軟體之指令，因此引入外製圖形時，T_EX 只是在版面上讓出一點空間，不作任何排版動作。排版之後，圖形在顯示器上出現或印出於白紙上必須藉助預覽/列印軟體。

市面上有各式各樣的繪圖系統，不同的系統可能使用不同的圖形語言。因為繪圖系統眾多，可以想見任何一套預覽/列印軟體不可能處理所有規格之圖形。本章主要介紹 PostScript 語言之圖形，並說明如何將此種圖形引入文稿中。此一圖形語言能力甚強，使用很廣。事實上，T_EX 排版系統與 PostScript 圖形語言之結合可以說是將排版品質推到一個新的高峰。

11.1 圖形檔案規格

簡單言之，要畫出一個黑白或彩色圖形有兩種方法，第一種是直接描點；第二種是先畫出外框，再把框內塗色。因此，圖形檔規格可以區分為兩大類：「描點式」(bit-mapped) 與「描邊式」(vector-based)。「描點式」圖形是一個黑點一個黑點地在白紙上畫出圖形來。Windows 內之 `imaging` 軟體所畫成的就是描點圖形。掃描器 (scanner) 所產生的也是描點式圖形。相反的，「描邊式」圖形則是描出圖形的外框，特定區域再塗上顏色。幾個有名的繪圖軟體，如 Corel Draw, Visio, Adobe Illustrator 等，所產生的都是描邊圖形。第 13.1.1 節 (頁 258) 所舉例之描點字形就是一個描點圖形；反之，描邊字形則是描邊圖形。

就使用彈性與排成品質而言，描邊圖形是最好的選擇。不過，描邊圖形本身也有許多種規格，其中 PostScript 是最廣泛使用的一種。PostScript 圖形語言雖然主要是用於處理描邊圖形，但它也有處理描點圖形的能力。舉例言之，我們用掃描器產生之圖形是描點格式。但透過工具軟體可以將它轉換為 PostScript 格式。轉換格式的好處是圖形在放大縮小時，較不會失真。本章主要介紹如何在文稿中引入 PostScript 圖形。不過，我們也將簡單說明如何處理描點圖形。有關於引用外製圖形的種種細節，請參考 [Reckdahl \(1997\)](#)。此一檔案對於引用外製圖形的各種問題有詳細說明，本章許多的介紹與說明即參考該檔案。

描邊圖形檔案在列印時可以放大、縮小、拉長或壓扁，而無損列印品質。而且，在不同精密度的印表機上印出，我們都可以得到同樣大小的圖形。反之，描點圖形在不同密度的印表機上列印，其大小會隨之而變。其次，圖形若放大或縮小，品質會變差。

列印品質和印表機種類有關。目前的雷射印表機每吋可印300點或600點；噴墨式列表機之列印密度有300點（如 HP 機種）也有360點（如 Canon 或 Epson 機種）。列印密度越高，表示它所列印的每一圓點比較小。說得更具體一點，600點之雷射印表機印出來的一個圓點，其直徑只有300點印表機的一半。高解析度印表機的圓點較小，因此能夠較精細地描繪圖形。反過來說，這也表示要描繪一固定大小之圖形，必須使用較多的圓點。

想像一個描點方塊，橫寬由900個黑白點，垂直由600個黑白點構成。此一圖形若由300dpi（每吋300點）雷射印表機印出，水平寬度為3吋，垂直高度為2吋。但是，同樣圖形如果由600dpi印表機印出，圖形變小，寬度縮小為1.5吋，高度變成1吋。因此，如果使用兩種不同密度的印表機，低密度印表機用於列印校對用初稿，高密度印表機用於印出最後完稿，則同一幅描點圖形就必須準備兩份檔案。否則初稿上圖形的大小將與最終完稿之大小不同。這是使用描點圖形不方便之處。

描邊圖形可以避免上述的缺點。不過，描邊圖形的規格有很多種，我們主要介紹的是 PostScript 圖形規格；這是美國 Adobe 公司所發展，因其能力特強，在1980年代迅速普及。目前，從300dpi的雷射印表機至2540dpi的相紙輸出機（phototypesetter），幾乎都能使用此一圖形語言。所謂印表機能使用 PostScript 語言，意思是說當電腦把一 PostScript 圖形檔傳送給印表機時，印表機有能力將它轉換圖形元素，在紙上印出。

市面的印表機一般是以品牌區分，但更重要的差別其實是其所使用的圖形語言。HP 相容型印表機使用的圖形語言稱為 PCL；Epson 或 Canon 噴墨印表機則使用另一種圖形語言。以上這些印表機若加上適當的硬體配備，就具有列印 PostScript 圖形的能力。具有列印 PostScript 圖形能力之印表機即稱為 PostScript 印表機，或簡稱為 PS 印表機。PS 印表機雖然功能較強，但價錢較貴。

幸運的是，網路上有一免費軟體 Ghostscript，可以讓一般的雷射印表機或噴墨印表機具備 PostScript 能力。Ghostscript 軟體是由 L. Peter Deutsch 所發展，它已經是專業排版者不可或缺的工具。假設我們有一 PostScript 圖形，但印表機是普通的 HP 或 Epson 印表機；則透過 Ghostscript 軟體可將此圖形由 HP 或 Epson 印表機印出。除了列印之外，Ghostscript 也可以將 PostScript 圖形顯示於電腦螢幕上。一般使用時，我們通常是透過 GSview 軟體執行 Ghostscript 程式。GSview 是由 Russell Lang 所發展。

11.2 引用外製圖形

PostScript 與 $\text{T}_{\text{E}}\text{X}$ 的功能很類似；只不過前者用於畫圖形，後者用於排版文字。如果熟悉 PostScript 圖形語言，我們也可以自行下指令畫出圖形來。但是，如果圖形複雜，直接下指令並不容易。一般的畫圖軟體，如 Corel Draw, Adobe Illustrator 等，主要的功能就是提供一使用者介面，讓我們可以輕鬆地在視窗內以滑鼠畫出圖形。當我們把圖形儲存（或輸出）為 PostScript 格式時，繪圖軟體即產生一 PostScript 檔案。

從排版的角度來看，我們並不是要畫一整頁的圖形，而是要將 PostScript 圖形引入版面特定位置。為達到此一目的，我們須使用一特別的格式，稱為 Encapsulated PostScript，通常簡稱為 EPS。相對而言，一般的 PostScript 圖形則簡稱為 PS。EPS 與一般的 PostScript 檔案其實很類似，主要的差別是前者檔案內須儲存有標示圖形大小的座標，底下將進一步說明。

在網路時代，JPEG 與 PDF 格式之圖形檔日漸普及。如果你要引用這兩類之圖形檔，可考慮使用 pdf $\text{T}_{\text{E}}\text{X}$ ，將排版文稿直接排成 PDF 格式，請參見 15.5.3 節（頁 306）之說明。

使用外製圖形的最大困難不在於如何下指令控制，而是在於產生正確的 EPS 檔案。很多圖形轉體所產生的 EPS 圖形檔都不太正確，引入文稿中即出現問題。

因此，你應使用品質可靠的繪圖軟體。另外，若文稿引用 EPS 圖形，最好是以 GSview 預覽或列印。Yap/Windvi 雖然也可以預覽或列印，但功能較差。本節首先說明引用外製圖形的指令，下一節再介紹幾種繪製 EPS 圖形的方法。

11.2.1 引用 PostScript 圖形

TeX 提供 `graphicx` 巨集套件以方便引用外製圖形。此一巨集套件除了可用以引入 EPS 圖形外，還可以引入 `pcx`, `tiff` 等格式之描點圖形。不過，底下之介紹主要是針對 EPS 圖形。

若圖形檔名為 `rice.eps`，引用圖形之指令如下：

```
\usepackage{graphicx}
\begin{figure}
\begin{center}
\includegraphics{rice.eps}
\caption{ 台灣的米價與蔗價 }
\end{center}
\end{figure}
...
```

EPS 圖形是以 `\includegraphics` 指令引入，這是 `graphicx` 巨集套件所提供之指令。指令環境 `center` 使圖形居中編排。指令環境 `figure` 具有浮動版面之功能，讓圖形置放於版面適當位置；`\caption` 指令則用於排版標題。

本例之圖形是以原尺寸引入，實際引用圖形時，我們通常須將原圖放大或縮小。欲調整圖形大小，可使用下列指令選項：

<code>height</code>	設定圖形高度，
<code>totalheight</code>	設定圖形全高，
<code>width</code>	設定圖形的寬度，
<code>angle</code>	設定圖形旋轉 (反時鐘方向) 角度，
<code>scale</code>	設定圖形放大 (或縮小) 之倍數。

想像把每一個圖形包含在一長方形中，左下角稱為參考點 (reference point)。由參考點起，可以算出圖形的高度 (`height`)，寬度 (`width`) 與深度 (`depth`)。若圖形不旋轉，深度為零，而全高 (`totalheight`) 等於其高度。若圖形旋轉 -45 度，參考點仍為原來位置，但深度將大於零，高度也改變了；此時全高即為高度與深度之和。

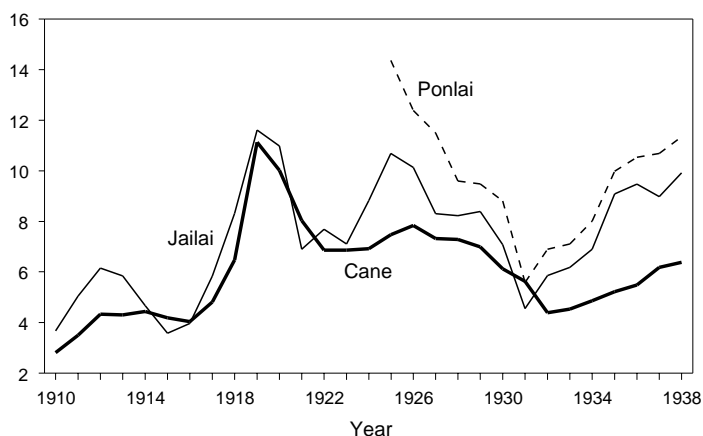


圖 11.1: 台灣的米價與蔗價

若要將圖形寬度拉大為與行寬相同, 指令如下:

```
\includegraphics[width=\textwidth]{rice.eps}
```

若圖形寬度要訂為行寬的 80%, 指令為:

```
\includegraphics[width=0.8\textwidth]{rice.eps}
```

如果要將圖形縮小為原尺寸的 80%, 須加上 [scale=0.8] 選項。在此選項下, 圖形橫寬與垂直高度將同比例變動。若要作不同比例的變動, 須同時選擇高度與寬度, 譬如:

```
\includegraphics[width=10cm,height=7cm]{rice.eps}
```

以上指令設定圖形高度為 7 公分, 寬度為 10 公分。除了以上之外, **graphicx** 巨集套件還有許多選項, 請參考其說明檔。

11.2.2 圖形中加入中文或數式

圖 11.1 的三條線是以英文標示其意義, 但在中文稿中, 圖形說明最好是使用中文。有時候, 我們須在圖中加上數學式, 如 $z = \sqrt{x^2 + y^2}$ 。在中文 Windows 的繪圖軟體中, 我們通常可以直接鍵入中文標題或說明文字。但是當圖形儲存

成描點格式的 `pcx` 或 `bmp` 檔案時，中文字不一定能正確儲存。如果把圖形儲存成 PostScript 格式，若繪圖軟體品質不佳，中文字部分也可能出現問題。同樣的問題也會出現在數學式或數學符號上。

解決以上問題的一個好方法是使用 `psfrag` 巨集套件。此巨集套件原由 Craig Barratt 所發展；3.0 版開始則由 Michael C. Grant 與 David Carlisle 共同發展。`psfrag` 巨集套件的原理很簡單，用軟體繪製圖形時，我們先加入英文標示，如圖 11.1 所示。引入圖形時，我們再下指令將英文標示字元轉換為中文或數學式。這個方法只能用於 PostScript 描邊圖形，描點圖形無法使用這個方法。

使用 `psfrag` 巨集套件之過程可以簡單描述如下：以軟體繪製圖形，於選定位置加入英文或數字標示；將圖形輸出為 EPS 格式之後再以 `\psfrag` 指令以新字元串替代原圖形之字元串。以圖 11.1 為例，原圖形內包含：Ponlai, Jailai, Cane, Year 及橫軸與縱軸上之阿拉伯數字。經過轉換之後，結果變成圖 11.2。底下進一步說明指令細節。

使用 `psfrag` 巨集套件，首先須於全文設定區引入。文稿內要將 Ponlai 英文字替代成爲「蓬來米」三個中文字，指令爲：

```
\psfrag{Ponlai}{ 蓬來米}
```

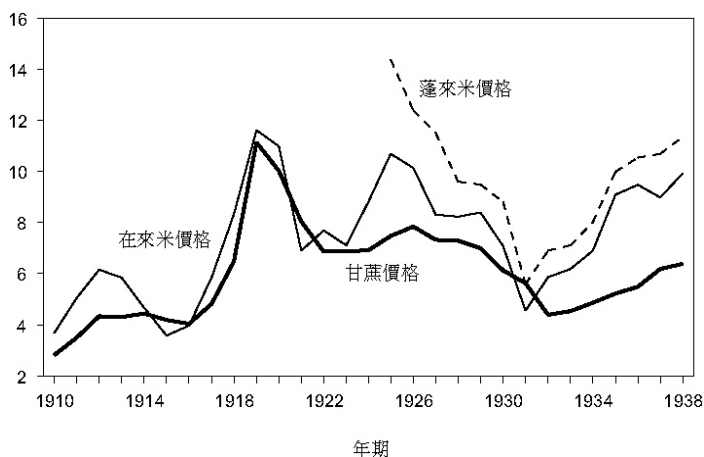
其中，替代之中文字體與大小可以任意選擇。另外，爲了讓替代字元置於適當位置，尚可加上調整位置之指令。想像原來圖中之 Ponlai 英文字與替代字元「蓬來米」三個中文字各有一長方形外框。每一外框可以英文字母 `t`, `b`, `l`, `r` 之組合設定個參考點：`t` 代表外框上方中央點，`b` 代表下方中央點，`l` 代表左方中央點，`r` 右方中央點。以上 4 個字母可以進一步組合，譬如，`t1` 代表長方形外框左上角，`bl` 爲左下角參考點，`b` 代表下方中間點；中心點則以 `□` 表示。

若要把替代字元之下方中心點置於原字元之上方中心點，指令爲：

```
\psfrag{Ponlai}[b][t]{ 蓬來米}
```

換言之，第一個方括號標示替代字元外框之參考位置，第二個方括號標示原始字元之外框參考位置。如果不標示位置，則兩串字元之參考位置將自動設爲 `[bl]`，亦即兩串字元外框之左下角將對準。

除了設定位置之外，替代字元還可以放大、縮小或旋轉。旋轉指令標示是指替代字元外框之參考點沿逆時針方向旋轉之角度。繼續以上的例子，若字體不放



```

\usepackage{psfrag,graphicx}
\begin{figure}
\psfrag{Year}[t][b]{  年期}
\psfrag{Ponlai}{  蓬來米價格}
\psfrag{Jilai}[r][r]{  在來米價格}
\psfrag{Cane}{  甘蔗價格}
\includegraphics[width=.8\textwidth]{rice.eps}
\end{figure}

```

圖 11.2: 圖形內之中文說明

大, 亦即放大倍數等於 1, 而字元要旋轉 90 度, 指令為:

```

\psfrag{Ponlai}[b][t][1][90]{  蓬來米}

```

其中, 第 3 對方括號中之數字表示放大倍數, 第 4 對方括號中之數字設定沿逆時針方向旋轉之角度。

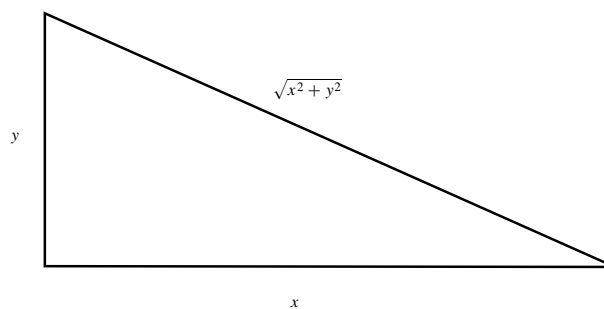
歸結以上的說明, `\psfrag` 指令之格式如下:

```

\psfrag{ ps}[pn][PSpn][scale][rot]{text}

```

其中, *ps* 表示原來圖形中之字元, *PSpn* 標示原字元外框之參考點, *pn* 標示替代字元之參考位置, *scale* 標示其放大倍數, *rot* 表示旋轉之角度, *text* 代表替代字串。圖 11.2 以 `\includegraphics{rice.eps}` 引入圖形; 在此指令之前, 替代字元則



```

\usepackage{psfrag,graphicx}
\begin{figure}
\psfrag{x}{$x$}
\psfrag{y}{$y$}
\psfrag{z}{$\sqrt{x^2+y^2}$}
\includegraphics[width=.8\textwidth]{triangle.eps}
\end{figure}

```

圖 11.3: 圖形中之數學式

以 `\psfrag` 指令引入。以第一行 `\psfrag` 指令為例，替代字元「年期」外框上方中心點對齊原字元 Year 外框下方中心點。

圖 11.3 是另外一個例子。我們先以繪圖軟體畫出三角形，底線以 x 標示，垂直線標示為 y ，斜線標示為 z 。繪圖完畢存檔之後，輸出為 EPS 格式，取名為 `triangle.eps`。輸出 EPS 檔應注意之細節，請見下一節之說明。本例引用圖形之指令中加入 `[width=.8\textwidth]` 選項，將圖形寬度調整為行寬的 80%。EPS 圖形的優點之一是圖形放大或縮小之後列印品質仍然很好。若是描點圖形，放大或縮小之後，列印品質通常不及原圖。

11.2.3 引用 PostScript 文稿檔案

本書第 2 章曾舉 7 個例子說明 \LaTeX 排版功能，每個例子之排版結果都同時引入版面中，以方便與輸入之指令對照比較。 \LaTeX 的排版結果可以轉換為 PostScript 檔案，若進一步轉換為 EPS 格式，即可引入另一篇文稿中。我們就是利用此種方法引入某些例子之排版結果。

欲將排版結果轉換為 EPS 格式，須注意 EPS 檔案僅能包含一頁版面。此外，EPS 檔案內若含有文字，轉換時應設定使用描邊字形，否則排版品質不佳。假設

原文稿檔名為 `example.ctx`，其中第 2 頁欲轉換為 EPS 格式。E_TE_X 排版完成之後，進一步執行下列指令：

```
c:\xtemp>dvips -E -p2 -l2 example.dvi
```

選項 `-E` 指示轉換為單頁之 EPS 格式；`-p2 -l2` 指示轉換第 2 頁。

使用 `-E` 選項時，DVIPS 程式會嘗試計算文稿版面之大小，並直接記錄於圖形檔案之 `BoundingBox` 指令內。但是，在某些情況下計算結果可能有誤。欲正確計算圖形版面之大小，可用下列方法：開啓 GSview 軟體，讀入 `example.ps` 圖形檔，啓動“File|PS to EPS”功能，確定視窗內之“Automatically calculate Bounding Box”已勾選，按一下 Yes 鍵之後，鍵入新檔名，可產生一新的 EPS 檔案，其內含正確的 Bounding Box 資訊。

以上說明產生 EPS 檔案的方法，不幸的是引用此圖形仍然有一個問題。將此一 EPS 檔案引入文稿，排版之後以 DVIPS 轉換為 `.ps` 檔案，以 GSview 預覽/列印時會出現錯誤。原因是目前版本的 DVIPS (5.86 版) 無法正確處理某些中文字型。但以 YAP/Windvi 預覽/列印則無問題。如果你購買了完整版本的 Acrobat 軟體，一個解決的方法如下：先利用 Acrobat Distiller 軟體將 EPS 檔案轉換為 PDF 格式；再開啓 Adobe Acrobat 軟體，讀入 PDF 檔案，將之轉換出 DVIPS 可以正確處理之 EPS 檔案。

11.3 繪製 EPS 圖形

產生 EPS 圖形檔案的方法之一是以繪圖軟體繪製。在 Windows 系統下，許多軟體都可以繪製圖形。譬如，Excel 試算表可由輸入資料自動畫出資料圖形；Corel Draw 或 Visio 軟體能繪製各式各樣圖形。此外，專業統計軟體如 Gauss，或者數學運算軟體如 Mathematica 等，也都能畫出各種圖形。某些繪圖軟體本身即提供直接產生 EPS 檔案的工具。例如以 Corel Draw 畫圖之後，可選擇將結果輸出為 EPS 檔案；但是，Excel 軟體並無此功能。

若軟體本身提供轉換工具，原則上應直接使用該項工具。不過，某些軟體雖然繪圖能力不錯，但轉換工具之功能卻有問題。此時，你可以將圖形剪貼到性能較佳的繪圖軟體中，再由該軟體輸出。如果繪圖軟體本身並無轉換功能，剪貼之後再輸出為 EPS 也是可能的途徑。譬如，Excel 軟體畫出之資料可以先剪貼到

Corel Draw 再輸出。不過，並不是所有人都有 Corel Draw 軟體。此時，還有一個辦法可以將圖形軟換為 EPS 檔案，那就是利用 PostScript 印表機驅動程式將圖形「列印」為 EPS 檔案。下一節將先說明如何安裝 PostScript 印表機驅動程式，之後再介紹轉換方法。

11.3.1 以 PostScript 驅動程式產生 EPS 圖形檔

Windows 作業系統提供 PostScript 印表機驅動程式。但是，Adobe 公司所提供之驅動程式功能最強，產生之 EPS 檔案最正確。若你要使用印表機驅動程式產生 EPS 檔案，應安裝此驅動程式。安裝程式名為 `winsteng.exe` 置於 `cwTeX` 光碟 `\support\adobe` 檔案夾下，直接執行即開始安裝。經過幾個選項畫面之後，會出現下列之選項畫面：

Available ports:

請點選 `File: Local ports` 。

安裝完成之後，進入 Windows 的「控制台」，點選「印表機」，除了原已安裝之印表機之外，會出現另一項 `Default PostScript Printer` ；列印 EPS 檔案時應選用此印表機。

利用 PostScript 列印驅動程式產生 EPS 圖形檔的方法很簡單，以下以 Excel 軟體為例說明之。畫 Excel 資料圖時最好把圖形置於獨立的一頁中。特別要注意是，圖形內之英文與數字不能選用中文細明體，必須改用 Windows 原有之英文字型如 Arial, Times New Roman 等。如果圖形內有中文字，驅動程式會把它們轉成描點字型，列印品質較差。因此，中文最好是以 `psfrag` 巨集套件處理。

資料圖畫好之後，先點選圖形區域，再將圖形印出。在接下來的「列印」畫面上，將「印表機」改為“`Generic PostScript Printer`”；並勾選「輸出至檔案」；按「確定」之後，填入自選的檔名，再按「確定」即產生 PostScript 檔案。一般 EPS 檔案是以 `eps` 為延伸檔名，但我們建議先以 `ps` 為延伸檔名。EPS 圖形檔與一般 PostScript 檔案不同的地方在於前者記錄有標示圖形大小的座標。不過，由驅動程式所轉換出來的檔案，其座標數字通常大於圖形實際大小。引入 `TeX` 文稿之後，圖形四周將出現多餘的空白。

簡單的解決辦法如下：以 `GSview` 軟體開啓 PostScript 檔案，以滑鼠點選 `File`，其下有“`PS to EPS`”選項；點選此一選項即可轉換出正確的 EPS 檔案。有時候，

你會發現 GSview 所讀取之圖形超過視窗所顯示的紙面範圍。此時，須回到 Excel 軟體中將圖形尺寸縮小一些，重新列印出 EPS 檔案。

11.3.2 使用繪圖軟體繪製 EPS 圖形

不同之軟體各有其繪製圖形的方法。一般的科學軟體，如 Mathematica 或 Gauss 等，都可以直接產生 PostScript 檔案。若使用 Corel Draw 或 Visio 繪圖軟體，我們直接在版面上拉出線條、上色彩、加文字。若使用 Windows 之 Excel 試算表軟體，我們先輸入數字，再由軟體畫出資料圖 (chart)。這種種方法所產生之圖形，都可以轉換為 PostScript 檔案。底下將以上述軟體為例，說明轉換檔案時須注意之細節。

以下所介紹之軟體大部分是商業繪圖軟體。商業軟體經常更新，各版本之功能不一定相容。而且，大部分軟體是國外公司所發展出版，因此在台灣可能同時流通中英文版本。一套軟體的不同版本性能可能有異；或者性能相同，但執行方式已改變。以下所介紹描述的主要是我們所熟悉使用者。如果你使用不同版本，甚或完全不同之軟體，你應該能從底下的說明中摸索出繪製 EPS 圖形的方法。

首先，我們說明繪圖時應注意之事項。大部分繪圖軟體都是由美加各國所開發，其設計主要是以英文使用者為對象。這些軟體經過中文化處理之後，我們可以在圖形中鍵入中文字，但是大部分軟體對於中英文之處理並不相同。簡單來說，大部分軟體都把中文字以描點圖形方式處理。相反的，如果你選用 Windows 內附之 Arial 或 Times New Roman 等英文字體，則圖形內之英文與數字大都以描邊字形處理。兩種處理方法的主要差別是在於列印品質的高低不同。如果繪製之圖形內並無中文，則英文與數字應選用 Windows 英文字體。

處理中文的主要考慮是品質。目前大部分軟體皆能直接鍵入中文字於圖形中，輸出之 EPS 檔案內也含有 (描點) 中文字。但是，最後之列印品質並不理想。使用 $\text{T}_\text{E}_\text{X}$ 排版的主要理由是追求品質。因此，底下所介紹的方法是以列印品質最佳為出發點。基本的技巧是透過 psfrag 巨集套件將中文字代入圖形中。

繪圖軟體: Corel Draw

在 Windows 下，我們可以選用之英文字型很多。若無特別偏好，英文與數字應選擇較常用之字體，如 Times New Roman 或 Arial；請勿使用中文字體 (如細明體或楷體) 之英文與數字。

畫好圖形，存檔之後，即可輸出 (Export) EPS 格式之檔案。以 Corel Draw 7.0 英文版為例，選定所欲輸出之圖形，點選 “File|Export”；「存檔類型」應選擇 Encapsulated PostScript (EPS) 。下一個畫面是 EPS Export ，有幾個選項須選擇。其中，Export text as 選項應選擇 Text ，並勾選 Include fonts 。EPS Export 畫面左邊第 2 欄位是 Image header ，其下之 Include header 請留為空白，勿勾選。若勾選此項，所產生之 EPS 圖形檔案前端會加入一段代表圖形之文字串。實際引用圖形時，可能會出現問題。

Export text as 選項下若勾選 Curves ，則圖形輸出為 EPS 時，圖形內之文字將轉換為一般之 PostScript 指令。反之，若選擇第 2 選項 Text ，圖形中之文字將直接存為字元，不作特別處理。就 T_EX 的排版應用而言，存為字元之圖形檔較小，而且接下來能夠以 psfrag 巨集套件作文字替換。

以上之說明是以 Corel Draw 英文 7.0 版為例；9.0 英文版也是以類似方法處理。若使用其他版本，作法可能稍有不同。而且，即使是同一版本號碼，中文版與英文版也可能不同，必須仔細測試才能確定。

繪圖軟體: Visio

繪圖軟體 Visio (4.0 版)，轉換檔案之基本原理與 Corel Draw 相同，但選項名稱稍有不同。在圖形繪製完畢開始輸出時，檔案格式應選擇 Encapsulated PostScript File。接下來的畫面有 Profiles ，Color Translation 及 Line Cap Mode 三選項。首先，Profiles 應選用 Standard Options-EPS ；其次，Color Translation 應選用 normal ；Line Cap Mode 可以選用 Device 。畫面下方另有 4 個選項：從 Background Rectangle 到 Include Preview ，應全部留為空白。

試算表軟體: Excel

Excel 所畫之資料圖可以利用 PostScript 印表機輸出為 EPS 檔案，也可以利用 Windows 之複製與貼下 (Copy and Paste) 功能，將圖形貼到繪圖軟體 Corel Draw 或 Visio 中，經過修改再輸出。自 Corel Draw 貼下時，須選用 Paste special 項下之 Picture (Metafile) 。相關之細節請見上文之說明。

Excel 圖形內若直接鍵入中文字，圖形內仍會出現。但中文字是以點陣方式 (bitmap) 儲存於圖檔內，圖形放大縮小時，中文字之列印品質較差。若要求較佳的列印結果，請使用 psfrag 巨集套件之方法。

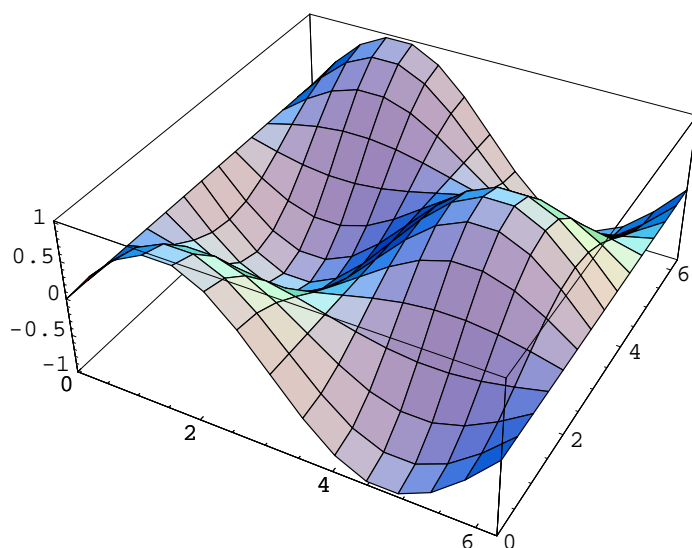


圖 11.4: 引用 Mathematica 軟體所產生的 EPS 圖形

計算軟體: Mathematica

Mathematica 是一個很有名的計算/繪圖軟體, 應用甚廣。底下以 3.0 版為例, 說明如何產生 EPS 圖形檔案。在 Mathematica 內執行下列指令:

```
Plot3D[Sin[x]Cos[y],{x,0,2 π},{y,0,2 π}]
```

螢幕上即畫出一彩色立體圖形。在圖形附近點選滑鼠左鍵, 將圖形框入於一長方形點線內。在圖框內按下滑鼠右鍵, 選取 Save Selection As 中之 EPS, 並鍵入圖形檔名, 硬碟內即產生 EPS 圖形檔。

以上所產生之圖形檔會把背景白紙包含在內, 若直接引入文稿內, 預覽排版結果時, 圖形變成很小, 或者版面一閃而過。解決辦法是在圖形檔案中加入正確的 bounding box。方法如下: 開啓 GSview 軟體, 讀入圖形檔, 啓動 “File|PS to EPS” 功能, 確定視窗內之 “Automatically calculate Bounding Box” 已勾選 (內定值), 按一下 Yes 鍵之後, 鍵入新檔名, 可產生一新的 EPS 檔案, 內含正確的 Bounding Box 座標。圖 11.4 是排版後之結果。

統計軟體: Gauss

Gauss 是有名的統計計量軟體, 亦可繪製 PostScript 圖形。以 MSDOS 系統下之 3.2.12 版為例, 將圖形檔轉換為 EPS 格式之步驟如下: 當圖形已出現於顯示器上, 請按朝上之箭頭鍵 (代表輸出圖形), 再按 c 鍵 (代表檔案轉換), 最後再按 e 鍵, 即可轉出 EPS 檔案。依原始設定, EPS 檔名將存為 `pgcvt.out`。如果要自動存為自己選擇之檔名, 請自硬碟中找出 `pggrun.cfg` 設定檔案, 將 `"cvt_filename="` 設定為自取之檔名。

畫圖時, 格點請勿設太細, 否則轉換出來之 EPS 檔案會很大, 預視或列印時速度會變慢。Gauss 為英文軟體, 無法直接鍵入中文字若要在圖中加入中文或數式, 請使用 `psfrag` 巨集套件。

11.3.3 引用圖形應注意之問題

引用圖形與使用 `psfrag` 巨集套件時, 請注意底下的問題:

- 圖形中所有英文與數字須選用英文字體, 如 Times New Roman。若以中文字體, 如新細明體或楷體, 書寫英文字母或數字, 所輸出之 EPS 檔案無法替換為中文字。
- 輸出 EPS 檔案時, 圖形中之英文與數字必須存為字元。以 Corel Draw 軟體為例, 輸出 (Export) 圖形時, 在 “Export text as ” 處有 “Curves ” 與 “Text ” 兩選項, 應選擇後者; 並勾選 “Include Fonts ”。
- 排版結果請以 GSview 預覽或列印, YAP/Windvi 無法顯示正確結果。
- 部分軟體輸出 EPS 檔案時會將字元串拆為兩行。譬如, Corel Draw 軟體有時會將一文字串拆成兩三行。因此, Year 可能拆成 Ye 與 ar 兩段; 或者 Yea 與 r 兩段。此時, 字元替代即無法成功。如果不能確定是否正確, 可用文字輸入軟體檢查 EPS 檔案, 看能否搜尋到原文字串。EPS 檔案是以普通文字儲存, 因此可以用文字編輯軟體修改。萬一真的拆成兩行或三行, 怎麼辦呢? 只好回到原軟體中, 將字串縮短, 期望下次轉換出正確結果。另外, Visio 繪圖軟體 (4.01 版) 輸出 EPS 圖形檔時, 各字元串後面會自動加上一控制指令: `\r`。引用圖形檔之前, 須先以文字編輯軟體將此控制指令去掉。
- 由 Excel 所轉換的資料圖中可能含有百分比符號 %。此符號在 \TeX 中是註銷指令, 出現在其後的文字將被忽略。因此, 被替代之字串中不可用 %。

- 下 `\psfrag` 指令時，須確定指令中之字串與 EPS 檔案內之字串完全相同，包括大小寫、空白等。
- 替代之文字並不限於是幾個文字，也可以是整段文字。此時，整段文字應置於 `\parbox` 或 `minipage` 指令環境中。
- 有些軟體雖然可以在圖形中加上英文或數字，但存檔時是以外框圖形儲存，而非字元形式。統計軟體 Gauss 就是一個例子。此時我們可以將畫好的圖形先引入畫圖軟體如 Corel Draw 或 Visio 中，加入適當文字之後，重新輸出為 EPS 檔案。

11.4 引用描點圖形

描點圖形格式眾多，最常用的包括 `tiff`，`pcx`，`bmp` 等等。例如，掃描器所產生之圖形通常儲存為 `tiff` 格式；Windows 系統內 `imaging` 軟體所繪出之圖形可儲存為 `pcx` 或 `bmp` 格式。描點圖形有固定解析度。例如，300dpi 掃描器通常產生 300dpi 之描點圖形。因為解析度固定，使用上並不方便。同一描點圖形在 600dpi 印表機印出時，圖形大小僅為 300dpi 印表機印出的 1/4。

描點圖形要引入文稿中，較方便的方法還是先把圖形轉換為 EPS 格式。依上一節的說明，利用 PostScript 印表機驅動程式可在 Windows 繪圖軟體內把圖形「列印」為 PostScript 檔案；其次利用 `GSview` 軟體再一次轉換為 EPS 檔案，以得到正確的圖形尺寸。圖 11.5 之例子原先是以掃描器掃描一張老照片，取得 `school.tif` 描點圖形，再依上述方法轉換為 `school.eps`，最後再引入文稿內。引用圖形之指令如下：

```
\usepackage{graphicx}
\renewcommand{\figurename}{ 圖}
\begin{figure}
\centering
\includegraphics[width=\textwidth]{school.pdf}
\caption{ 台北高等商業學校 (今日台大社會科學/法學院)}
\end{figure}
```

11.5 圖形位置與標題

圖 11.5 的例子使用 `\caption` 指令排版標題。此一指令僅能用於 `table` 與 `figure`



圖 11.5: 台北高等商業學校 (今日台大社會科學/法學院)

指令環境內; 在 \LaTeX 中這兩者合稱為「浮動版面」指令環境。10.4 節 (頁 204) 曾說明如何利用 `table` 指令環境控制表格位置, `figure` 指令環境之功能與 `table` 指令環境相同, 唯一的差別只是在於標題之名稱不同。在 `figure` 指令環境內, 使用 `\caption` 指令排版時, 圖編號之前會自動加上 “Figure” 名稱。若排版中文稿件, 如本例所示我們可將 `\figurename` 改為「圖」或其他中文字, 。

12 圖形與彩色

上一章說明引用外製圖形的方法。巨集套件 `graphicx` 除了可引進外製圖形外，還可旋轉文字、圖表等。另外，`color` 巨集套件可以將圖形、文字上色彩，或加上灰階 (grayscale) 之背景方塊。若圖形不複雜，我們也可以使用 `picture` 指令環境中之指令直接繪製圖形於文稿中。

\TeX 的繪圖功能有限，大部分的繪圖功能必須透過預覽/列印軟體來執行。簡單來說，文稿中若引用外製圖形，排版時 \TeX 只在適當地方留出圖形大小的空白。等到列印或預視時，圖形才由預覽/列印軟體引進版面上。本章將介紹與圖形處理有關的一些巨集套件。若使用這些巨集套件，通常排版文稿須先以 `DVIPS` 程式轉換為 `PostScript` 檔案，再以 `GSview` 預覽/列印。若直接以 `YAP/Windvi` 預覽/列印，有時候無法得到正確的結果。

12.1 `graphicx` 巨集套件

\TeX 提供 `graphicx` 及 `graphics` 兩套圖形巨集指令。兩套巨集套件之功能相近，但前者使用較方便，故本章僅介紹前者之指令。

巨集套件 `graphicx` 之指令可用以引入外製圖形，旋轉或縮放文字圖表。如果是以 `DVIPS` 程式將排版結果轉換為 `PostScript` 檔案，則使用 `graphicx` 巨集套件時，應加入選項：

```
\usepackage[dvips]{graphicx}
```

另外一個辦法是建立 `graphics.cfg` 檔案，置於 `c:\texmf\tex\latex\config` 檔案夾內；檔案只須包含一行指令：

```
\ExecuteOptions{dvips}
```

若建有此檔案，引用巨集套件時可以省略 `[dvips]` 選項。`MiKTeX` 與 `fpTeX` 系統都已自動建立設定檔案，使用者不須再費心。

巨集套件 `graphicx` 的第一項功能是引入外製圖形，上一章對此已有詳細的說明。除此之外，`graphicx` 之指令可用來旋轉或放大縮小文字、圖表。以下分別介紹之。

12.1.1 旋轉文字圖表

有時候，我們須將版面上部分圖文旋轉某一角度。譬如，若表格橫寬太大，須旋轉 90 度才能排入版面，此時可使用 `\rotatebox` 指令，其格式如下：

$$\backslash\text{rotatebox}[\text{options}]\{\text{angle}\}\{\text{material}\}$$

其中，`options` 表選項，`angle` 表旋轉角度，`material` 表欲旋轉之文字或圖表。若文字段落超過一行，應將文字段落置於 `\parbox` 或 `minipage` 指令環境中。應注意的是，全部文字應置於一個段落內。換言之，欲旋轉之文字或表格中，不得空一行或加上 `\par` 指令。但是換行 `\\` 指令可以使用。

旋轉角度是以逆時鐘方向計算；旋轉時以基準點 (reference point) 為軸心。若為單一文字，基準點是基線與左外框之交點；若為圖表，基準點為左下角。若要以特定点為旋轉軸心，我們可以經由選項 `origin=` 來控制。選項設定可以選擇 `c`, `l`, `r`, `t`, `b`, `B` 分別代表文字圖表的中心點，左，右，上，下，基線。因此，若以圖形中心點為旋轉軸心，選項指令為 `origin=c`。上述之設定項還可加以組合，例如 `lb` 代表旋轉軸心為左下角。若有必要，我們還可以直接選擇軸心之座標。細節請見 `graphicx` 巨集套件之說明檔，或 [Goossens et al. \(1997\)](#)。

圖 12.1 之例子說明不同旋轉軸心點之效果。例子中，我們先定義排版 `cwTeX` 標誌符號之指令。為了方便比較，每一行排出三個標誌符號；前後兩個不作旋轉，中間的符號則旋轉 15 度。但依旋轉軸心不同，排版結果也有明顯差異。第 1 行指令選擇以圖形中心點旋轉；第 2 行未設定選項，因此以基準點為軸心；第 3 行直接標示軸心點位置距離基點 (20pt,10pt)。

圖 12.2 是另一個旋轉文字圖表的例子。例子上半部分先排版一簡單表格，其中使用 `picture` 指令環境之 `\line` 與 `\put` 指令畫一斜線。有關於 `picture` 指令環境之畫圖指令的使用方法，請見下一節之說明。例子下半部分說明如何將表格旋轉 90 度。旋轉圖表之指令為 `\rotatebox`，但首先須引入 `graphicx` 巨集套件；旋轉軸心設定為原表格之中心點。

$cwTeX \quad cwTeX \quad cwTeX$ $cwTeX \quad cwTeX \quad cwTeX$ $cwTeX \quad cwTeX \quad cwTeX$	<pre> \usepackage{graphicx} \def\cw{\tt cw}\TeX{} \cw \rotatebox[origin=c]{% {15}{\cw} \cw\ \cw \rotatebox{15}{\cw} \cw\ \cw \rotatebox[x=20pt,y=10pt]{% {15}{\cw} \cw </pre>
-------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

圖 12.1: 旋轉文字圖表例1

<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 5px;">$x \backslash y$</td> <td style="padding: 5px;">y_1</td> <td style="padding: 5px;">y_2</td> </tr> <tr> <td style="padding: 5px;">1</td> <td style="padding: 5px;">34</td> <td style="padding: 5px;">55</td> </tr> <tr> <td style="padding: 5px;">2</td> <td style="padding: 5px;">25</td> <td style="padding: 5px;">45</td> </tr> </table> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 5px;">y_2</td> <td style="padding: 5px;">34</td> <td style="padding: 5px;">55</td> </tr> <tr> <td style="padding: 5px;">y_1</td> <td style="padding: 5px;">25</td> <td style="padding: 5px;">45</td> </tr> <tr> <td style="padding: 5px;">$y \backslash x$</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">2</td> </tr> </table>	$x \backslash y$	y_1	y_2	1	34	55	2	25	45	y_2	34	55	y_1	25	45	$y \backslash x$	1	2	<pre> \begin{tabular}{l@{r}rr} & \mathit{y} & & \\ \mathit{x} \backslash & \mathit{y}_1 & \mathit{y}_2 & \\ 1 & 34 & 55 & \\ 2 & 25 & 45 & \\ \hline 1 & 34 & 55 & \\ 2 & 25 & 45 & \\ \end{tabular} \usepackage{graphicx} \rotatebox[origin=c]{90}{ \begin{tabular}{l@{r}rr} ... \end{tabular}} </pre>
$x \backslash y$	y_1	y_2																	
1	34	55																	
2	25	45																	
y_2	34	55																	
y_1	25	45																	
$y \backslash x$	1	2																	

圖 12.2: 旋轉文字圖表例2

12.1.2 放大或縮小文字圖表

$cwTeX$ 安裝程式設定排版時使用描邊字型, 此一設定的優點是文字可以任意放大縮小, 品質不受影響。就圖形部分而言, 若以 `\includegraphics` 指令引入之圖形為 EPS 格式, 我們也可以設定任意放大或縮小, 而無損品質。

不過, 如果是要作不同比例之放大, 則使用 `graphicx` 巨集套件所提供之指令還是較方便。欲放大與縮小文字圖表, 可使用下列指令:

```

\scalebox{ scalefact}{material}
\resizebox*{ h-scale}{v-vscalescale}{material}
\reflectbox{ material}

```

第一道指令 `\scalebox` 是用於將文字圖形比例放大為選定之倍數, `scalefact` 數字選項設定放大倍數。放大或縮小時, 水平與垂直方向是同一比例。若水平與垂

直方向要以不同比例放大或縮小，則原設定倍數為水平放大倍數；其後須加上垂直放大倍數選項。例如，`\scalebox{2}[1.5]{ text}` 表水平倍數為 2，垂直倍數為 1.5。請注意，垂直倍數是置於方括號內，請見以下例子：

$\text{cwT}_{\text{E}}\text{X}$ $\text{cwT}_{\text{E}}\text{X}$	<pre>\usepackage{graphicx} \scalebox{2}{\cw}\ \scalebox{1.2}[1.4]{\cw}</pre>
-----------------------------------------------------------------	------------------------------------------------------------------------------

若要放大或縮小為選定之尺寸，應使用 `\resizebox*` 或 `\resizebox` 指令。例如，`\resizebox*{5cm}{4cm}{...}` 可將文字圖表轉變成 5 公分寬，4 公分高。若使用無 * 號指令格式，則計算放大倍數時，只考慮原文字圖表之高度 (height)，深度 (depth) 不列入考慮。一般而言，使用加上 * 之指令格式較方便。

若要將文字圖表之寬度拉大至特定大小，如 5 公分，而高度希望同比例放大，可以使用下列指令：`\resizebox*{5cm}{!}`。請參見底下的例子。

$\text{cwT}_{\text{E}}\text{X}$	<pre>\resizebox*{2cm}{!}{\cw}</pre>
---------------------------------	-------------------------------------

`\scalebox` 指令之水平倍數若為負值，結果為鏡面反映。因此，`\reflectbox` 指令之效果與 `\scalebox{-1}[1]` 完全相同。

$X_{\text{q}}\Gamma_{\text{w}}$ $X_{\text{q}}\Gamma_{\text{w}}$	<pre>\reflectbox{\cw} \ \scalebox{-1}[1]{\cw}</pre>
-----------------------------------------------------------------	-----------------------------------------------------

12.2 picture 指令環境

要在文稿中直接繪製簡單線條圖形，可使用 `picture` 指令環境，或者透過巨集套件間接引用 PostScript 或 METAFONT 之繪圖指令。如果只是簡單的線條圖形，`picture` 指令通常就能完成使命。使用 `picture` 指令的好處是圖形指令直接下於文稿中，不須另存圖形檔。本節將簡單介紹 `picture` 指令環境之功能。

圖形占有一定之空白，繪製圖形時首先須在版面上留下特定大小之空白。一般圖形所占用之空白為長方形，其大小以座標點表示。例如，寬度 100、高度 60 之

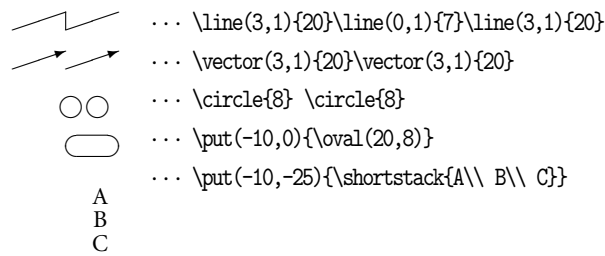


圖 12.3: picture 指令環境

長方形,可定為左下角座標為 (0,0), 右上角為 (100,60)。排版時,長方形之左下角將排於基線上。座標的單位長度可以 `\unitlength` 指令任意設定, 例如:

```
\setlength{\unitlength}{mm}
```

即選定公厘 (mm) 為長度單位。如此, 上例之長方形寬度為 10 公分, 高度為 6 公分。長方形之左下角位置可以進一步設定, 例如以下指令所定義之長方形其左下角之位置距離原基準點 (20mm,-10mm) :

```
\setlength{\unitlength}{mm}
\begin{picture}(100,60)(20,-10)
...
\end{picture}
```

在 `picture` 指令環境下, 畫直線之指令為 `\line`。譬如 `\line(2,3){10}` 表示畫一直線, 起始點為 (0,0), 方向為 (2,3); 長度為 10。進一步利用 `\put` 指令, 即可將此直線移至版面上選定的位置。畫箭號之指令為 `\vector`, 圓圈之指令為 `\circle`, 橢圓之指令為 `\oval`。圖 12.3 列出幾個簡單的例子。

我們將 `picture` 指令環境內之畫圖指令簡單歸納如下:

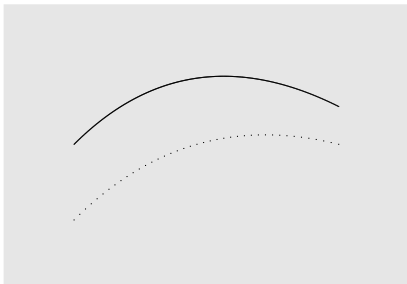
- `\put(x,y){...}` : 將繪製之線條、方塊文字等置於 (x,y) 座標處。
- `\multiput(x,y)(a,b){n}{...}` : 將繪製之圖形文字或線條等重覆排版 n 次, 起始位置之座標為 (x,y), 下一個位置之座標為 (x+a,y+b), 餘此類推。譬如, 底下指令 `\multiput(0,0)(10,1){3}{\bullet}` 產生: ● ● ●
- `\line(x,y){ length}`: 繪製直線, 起始點為 (0,0), 角度方向為 (x,y), 長度為 length。參見上面例子之第一行。特別注意的是, 設定角度方向之座標

有下列限制: (1) 座標值必須是正負整數, 不可使用小數; (2) 數值只能介於 -6 與 6 之間; (3) x, y 座標值不能有公約數。譬如, (3,6) 不能使用, 因為兩數可以 3 整除, 但 (1,2) 則無問題。

- `\vector(x,y){ length}`: 類似 `\line` 指令, 但前端加上箭頭。
- `\circle{x}` : 以基準點為圓心, x 為半徑畫圓。
- `\oval{x,y}` : 畫橢圓, 寬度為 x , 高度為 y 。橢圓可以想像為是一長方形, 大小為 (x,y) ; 其四角改為弧形之後即變為橢圓。
- `\qbezier[n](x1,y1)(x2,y2)(x3,y3)` : 畫 bezier 曲線。選項 $[n]$ 若省略, 結果為實曲線; 若加入, 為虛曲線。下面將舉一例子說明。
- `\shortstack` : 將文字圖表垂直疊在一起, 各行以斷行指令隔開。
- `\thicklines` , `\thinlines` : 設定指令環境內線條之粗細。後者為內設值。此外, 我們尚可直接設定粗細。例如, `\linethickness{2pt}` 指令即設定線條粗細為 2pt。
- `\dashbox{w}(x,y)[pos]{...}` : 將圖表加上點折線之方形外框。其中, $\{w\}$ 選項設定每一點折線之長度。 (x,y) 設定長方形外框之尺寸, $[pos]$ 設定文字圖表位於框內之位置。 $[pos]$ 可選用 `t`, `b`, `l`, `r`, 及 `s`。前四選項分別代表文字圖表位於框內靠上, 靠下方, 靠左, 靠右。以上之選項可進一步組合, 譬如 `[t1]` 表靠左上角。 `s` 選項則表示圖表在水平方向將選定之外框填滿, 垂直方向則居中。此外, `\framebox` 與 `\makebox` 也可以在 `picture` 指令環境內使用。

以上所介紹之指令大部分用於畫直線或圓形, 但我們可以使用 `\qbezier` 指令畫曲線圖, 圖 12.4 是一個簡單的例子。此項指令須設定 3 個座標, 第 1 個座標為曲線起點, 第 2 個座標為線條前進之方向, 第 3 個座標為經過中間轉折之後, 線條所抵達的終點。第二道指令之後加上 `[40]` 之選項, 整條曲線變成由 40 個細點所構成的曲線。

以上所介紹的 `picture` 指令雖然功能簡單, 但用於繪製簡單之線條圖形卻是綽綽有餘。不過, 繪圖時須設定每一線條之位置、長度等等, 使用上不是很方便。有一些巨集套件以上述指令為基礎, 但讓使用者方便繪製圖形。譬如, Peter Vanroose 之 `trees` 巨集套件用於繪製樹狀圖甚為方便。



```
\setlength{\unitlength}{1mm}
\begin{picture}(40,30)
\qbezier(5,15)(20,30)(40,20)
\qbezier[40](5,5)(20,20)(40,15)
\end{picture}
```

圖 12.4: 曲線圖

另外一些巨集套件則是延伸 `picture` 指令環境之功能。譬如, Sunil Podar 之 `epic`, 或者 Conrad Kwok 之 `eepic` 巨集套件都屬於此類。以上之巨集套件可以自網路上取得, 其中內附有說明檔。使用者亦可參考 [Goossens et al. \(1994\)](#) 與 [Goossens et al. \(1997\)](#) 之比較說明。

12.3 彩色圖文

欲將文字或圖形加上色彩, 最簡單的方法是使用 \TeX 所提供之 `color` 巨集套件。本節所謂的「色彩」, 包括不同層次之灰階 (grayscale)。使用 `color` 巨集套件時, 必須配合以特定之列印與視覽程式。最簡單的方法是以 `DVIPS` 程式產生 `PostScript` 檔案, 再以 `GSview` 預覽/列印。如果以 `YAP` 或 `Windvi` 有些顏色可能無法正確顯示或列印。

色彩巨集套件最常見之應用有兩類, 一是將文字著色, 一是在文字或圖形之後加上灰階或顏色背景。`color` 巨集套件提供一些事先定義之顏色, 但也可以自行定義所要的顏色。使用色彩指令, 首先在全文設定區引入 `color` 巨集套件:

```
\usepackage[dvips]{color}
```

其中, `[dvips]` 選項的目的是設定使用 `DVIPS` 程式轉換排版結果。`MiKTeX` 與 `fpTeX` 都已自動設定, 故此選項可以不加。

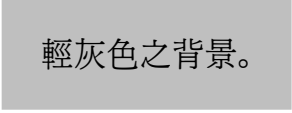
欲將文字塗上色彩, 指令如下:

```
\textcolor{red}{\ctxfbb 請注意}, 三分鐘之後 ...
```

排版之後, 以 `GSview` 軟體預視, 「請注意」三個字將為紅色之粗黑體字。可以直

接使用之色彩選項為 `black` , `white` , `red` , `green` , `blue` , `yellow` , `cyan` 與 `magenta` 。若要自行調色, 請參考 `color` 巨集套件之說明, 或者 [Goossens et al. \(1997\)](#)。

欲將文字或圖形加上灰階背景, 可以使用 `\colorbox` 指令。使用灰階色之前, 首先須定義灰階色。灰階色深淺是以一介於0與1之間的數字代表, 指數為1表示全白, 指數0表示全黑。底下的指令定義 `slight` 為0.75度之灰階色, 再以之創造之灰階方塊:



輕灰色之背景。

```
\usepackage{color}
\fbxsep=15pt
\definecolor{slight}{gray}{0.75}
\colorbox{slight}{\Large 輕灰色之背景。}
```

第2行指令定義灰階之深淺, `gray` 是 `color` 巨集套件之指令, `slight` 則是自行選定之顏色名稱。另外, 背景外框與文字之距離可由 `\fbxsep` 指令控制, 本例子之設定為15pt。

若要将整段文字加上灰階背景, 可將整段文字排於 `minipage` 指令環境內, 再以顏色指令上色。例如:

```
\colorbox{light}{\begin{minipage}
...
\end{minipage}}
```

如果要將整頁版面填上顏色, 可使用 `\pagecolor` 指令。

底下例子以 `\textcolor` 與 `\colorbox` 兩項指令創造出黑底白字之效果。其中, `\colorbox` 指令先創造出一接近黑色之方塊, 再以 `\textcolor` 指令在其中「挖出」白字:



請特別注意！

```
\usepackage{color}
\fbxsep=15pt
\definecolor{dark}{gray}{0.4}
\colorbox{dark}{\textcolor{white}%
{\LARGE \ctxfbb 請特別注意 !}}
```

指令第4行之 `white` 代表純白色, 這是 `color` 巨集套件所內定之現成顏色, 故不須再自行定義。如上所述, 除了白色之外, 尚有其他內定之顏色, 如 `red` , `blue` 等, 都可以直接引用。此外, 本例子也使用 `\fbxsep=15pt` 指令控制方格邊緣與其內

文字之間距。

12.4 lettrine 巨集套件

在 洋文書中，我們常見到將章節起頭第一個字母特別放大，有時候占用兩行或三行的高度。此種排版稱為 drop caps。Daniel Flipo 所寫的 `lettrine` 巨集套件可用於排版 drop caps。本段第一個中文字之排版指令如下：

```
\lettrine[lines=2]{ 在 } 洋文書中 ...
```

指令第 1 選項設定第一個字的高度要占多少行。若不加選項，內設值為 2 行。在英文稿中，一個單字是由多個字母組成，若僅第一個字母要放大，則將該字母置於第一對大括號內，其餘字母則置於第二對大括號內。

除了行數設定之外，還可加入其他選項調整單字位置。譬如，`lhang=0.3` 表示要將大字左移 30%，`loversize=0.1` 則表示要將單字放大到使其高度高於正常高度的 10%。

字 體放大之後，筆劃變粗。若以全黑印出，太過醒目，特別是筆劃較多的中文字更是如此。本段第一個字使用 `\textcolor` 指令設定以 40% 灰階排版。排版指令如下：

```
\lettrine[lines=2, lhang=0.2]{\textcolor{heavy}{\ctxfbb 字}} 體 ...
```

12.5 contour 巨集套件

較晚近的繪圖軟體大都提供創造立體陰影 (dropping shadows) 效果之指令。要在 \TeX 文稿中產生類似的效果，可使用 `contour` 巨集套件，作者為 Harald Harders。此一巨集套件的能力當然不能和一般的繪圖軟體相比，但在某些應用下，使用相當方便。

欲使用此巨集套件，首先須定義一彩色或灰階色。譬如，以下指令：

```
\usepackage{contour}
\contour{slight}{\Huge \ctxfbb 台大經濟系}
```

排版結果為：

台大經濟系

其中, `slight` 為 0.75 灰階度。若將 `slight` 改為其他灰階色, 排版效果會不一樣, 有興趣者請自行嘗試。

加上立體陰影之文字也可以置於灰階背景上, 例如:

```
\colorbox{slight}{\contour{white}{\ctxfbb 台大經濟系}}
```

排版結果為:

台大經濟系

巨集套件 `contour` 所產生的陰影效果尚可自行調整, 請參閱其說明檔。

12.6 wrapfig 巨集套件

引用外製圖形時, 通常是使用 `figure` 指令環境將圖表置於版面上方或下方。但有時候, 我們希望把圖形或表格置於正文方格內, 或者版面靠左邊, 或者靠右邊; 甚至是置於正方塊內, 四周以文字包圍。此時可以使用 Donald Arseneau 所寫的 `wrapfig` 巨集套件。表面上看來, `wrapfig` 之功能類似 `figure` 指令環境, 但指令與實際功能並不相同。

舉例言之, 版面旁邊的 EPS 圖形是以 `wrapfig` 指令環境引入, 使用之指令如下:



```
\begin{wrapfigure}[3]{1}[1cm]{3.4cm}
\includegraphics[width=3cm]{cat.pdf}
\end{wrapfigure}
\mbox{}
舉例言之, 版面 ...
```

以上指令是下於上段與本段文字之間, 其下立即接上本段內文。請注意, 在指令環境之後我們須特別加上 `\mbox{}` 指令, 否則排版時會出現錯誤。(其原因不明, 但可能與下接之文字為中文字有關。)

`wrapfig` 指令環境第 1 選項 [3] 設定圖形高度占 3 行文字。此選項若不加上, 巨集套件會自動計算。第 2 選項 {1} 指示將圖形置於版面左邊。若選項改為 {r}

則圖形將置於右邊。第3選項 [1cm] 指示將圖形凸出於版面邊緣外 1 公分。若不加入此項設定, 內設值為 [0cm]。第4選項 {3.4cm} 指示圖形所占寬度為 3.4 分。實際引入圖形時, 寬度設定為 3 公分, 讓左右兩邊各有一些空白。巨集指令會自動計算圖形本身所占之高度, 不過我們也可以自行設定其高度; 設定方法請見巨集套件內附之說明。

請注意, `wrapfigure` 指令環境之後須緊接著輸入段落文字, 否則會出現錯誤。上述指令第一行為 `\intextsep=Opt`, 其功能是設定圖形與周圍文字之間距。巨集套件 `wrapfig` 應用上有一些限制, 其使用說明直接置於 `wrapfig.sty` 檔案內, 請自行參考。

12.7 PSTricks 巨集套件

上一節說明如何在文稿中直接繪製線條圖形。但是, \TeX 的繪圖指令功能不強, 較複雜的圖形即無能為力。第 11 章曾說明如何引用外製圖形。 \TeX 引用 PS 檔案時, 實際上是引入一連串的 PostScript 指令。因此, 必要時我們也可以在文稿中直接使用 PostScript 指令。要在文稿中使用 PostScript 畫圖指令, Timothy van Zandt 所寫的 PSTricks 巨集套件可能是功能最強、指令最完整者。此一巨集套件內含完整的指令說明檔案。Goossens et al. (1997) 對此巨集套件有詳細的說明, 並與其他繪圖巨集套件作比較, 很值得參考。本節之說明主要參考該書。

PSTricks 功能強大, 指令很多, 我們無法在此詳細說明。底下只以幾個例子展示其功能, 欲了解這些例子所使用的指令, 請參考巨集套件之說明檔。PSTricks 套件指令直接使用 PS 指令, 因此我們必須使用 PS 印表機, 或者 GSview 軟體才能列印或預覽排版結果。

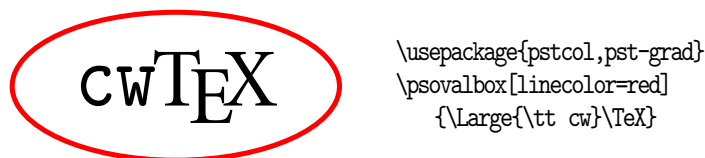
PSTricks 早在 1993 年就發展出來, 它和後來發展的 \TeX 有部分地方不完全匹配。如果你要同時使用 `graphicx`, `color` 與 PSTricks 巨集套件, 你還必須引入 David Carlisle 所寫的 `pstcol` 巨集套件, 而且各巨集套件應依以上順序引入。譬如, 若欲引用畫樹狀圖之巨集套件 `pst-tree`, 全文設定區所下之指令為:

```
\usepackage{graphicx,color,pstcol,pst-tree}
```

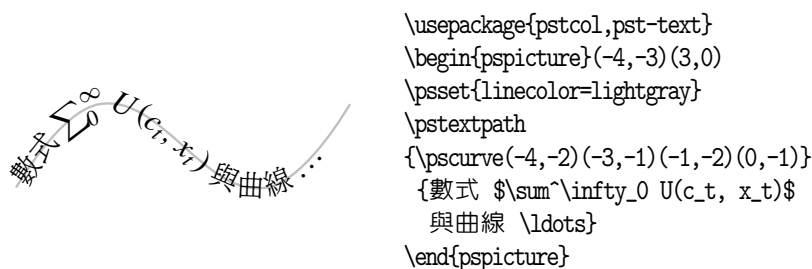
若引入順序不對, 排版時出現錯誤。

整套 PSTricks 巨集套件包含幾個較小的巨集套件, 各有不同功能, 底下以幾

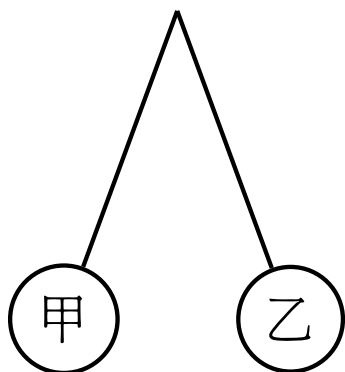
個例子展示其功能。第一個例子是將文字加入 PS 指令所繪成的橢圓中。指令中設定橢圓是以紅線畫出。我們還可以進一步使用指令讓 `cwTeX` 標誌符號以另一種顏色排版出來。若使用 `GSview` 軟體預視, 即可看到彩色的圖案; 使用彩色印表機即可印出紅色橢圓。以單色印表機列印時, 紅線自動轉換為灰階線條。



PostScript 處理文字的功能特別強。12.1 節 (頁 243) 曾介紹如何以 `graphicx` 巨集套件指令將文字圖形放大、縮小、旋轉等。`PSTricks` 巨集套件同樣有這些功能。而且, 後者的某些功能是前者無法做到的。最後一個例子先畫出一條灰階曲線, 再讓文字與數式沿著此一條線的形狀排列出來。

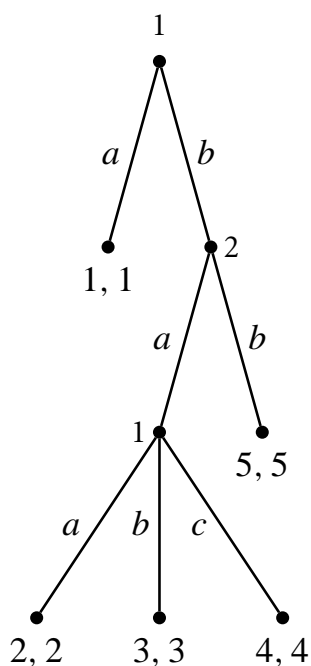


最後我們說明如何繪製樹狀圖 (trees)。許多學科領域都可能須繪製樹狀圖, 但在 `TeX` 文稿內直接繪製樹狀圖並不容易。`PSTricks` 繪製樹狀圖之能力甚例, 如果你經常須繪製樹狀圖, 則花一點時間了解畫圖指令, 絕對是一個值得的投資。首先, 我們介紹一個簡單的例子。繪製本圖形, 除了 `pstcol` 巨集套件之外, 尚需 `pst-node` 及 `pst-tree` 巨集套件。實際應用時, 樹狀圖可以一層一層地接續下去; 每一分叉處可以排版文字、粗黑點、圓圈、方塊等等。

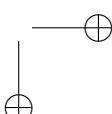
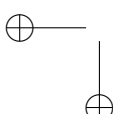
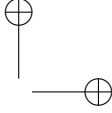
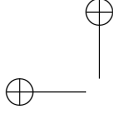


```
\usepackage{pstcol,pst-node,pst-tree}
\pstree{\Tp}{
\Tcircle{ 甲}
\Tcircle{ 乙}}
```

底下是一個較複雜的樹狀圖例子。



```
\usepackage{pst-node,pst-tree}
\psset{labelsep=2pt,tnpos=a,radius=2pt}
\pstree{\TC*~{1}}
{\TC*~{}~[tnpos=b]{$1,1$}\tput{$a$}
\pstree{\TC*~{}~[tnpos=r]{2}\trput{$b$}}
{\pstree{\TC*~{}~[tnpos=1]{1}\tput{$a$}}
{\TC*~{}~[tnpos=b]{$2,2$}\tput{$a$}
\TC*~{}~[tnpos=b]{$3,3$}\tput{$b$}
\TC*~{}~[tnpos=b]{$4,4$}\tput{$c$}}
\TC*~{}~[tnpos=b]{$5,5$}\trput{$b$}}
```



13 PostScript 字體

中國於第 11 世紀發明活字排版技術，德國人 Johannes Gutenberg 則於 15 世紀發明類似的技術。從 15 世紀至 20 世紀中葉，活字排版的技術基本上並無改變。傳統排版工廠裡，排版工人自字盤上取出鉛字，一個一個併在一起，組成一頁一頁的版面。字模沾上油墨，壓印於白紙上，即印出書稿。因此，排成品質的高低決定於兩項因素：一是字體本身是否優美，二是版面設計是否適當。

1980 年代初期，電腦排版開始發展；排版軟體取代了傳統的排版工人，電腦字體則取代了傳統的鉛字。從這個角度來看， $\text{T}_{\text{E}}\text{X}$ 排版系統分兩部分，第一部分是 $\text{T}_{\text{E}}\text{X}$ 軟體本身，其功能是版面安排；第二部分是各式各樣的英數字體。傳統的鉛字是將字體鑄於鉛塊上，電腦字體則是將文字的形狀記錄於程式裡。換言之，每一個單字的形狀，不管是中、英文，都是以電腦程式描繪其形狀。排版之後，列印程式解讀每一個字型程式，變成黑白或彩色細點，驅動印表機印出結果。在現代科技下，每一個字形都是一組程式。

但是，同樣一個英文字母卻有不同的程式寫法。譬如，Knuth 所使用之造字軟體稱為 METAFONT，其所創造出來的字體稱為 METAFONT 字體。另一套有名的字型規格是 Adobe 公司所發展的 PostScript 字體，又稱為 Type1 字體。雖然 $\text{T}_{\text{E}}\text{X}$ 一開始使用 METAFONT 字體，但以目前的發展狀況來看，PostScript 字體的使用彈性最大，字型選擇也最多。本章將簡單說明 $\text{E}_{\text{T}}\text{X}$ 如何使用 PostScript 字型排版，並介紹一些與使用字體有關的工具程式。

13.1 METAFONT 與 PostScript 字體

Knuth 設計了八十多種字體，其中有些是一般的英數字母，其餘是用於排版數學式的符號。大部分的字型檔案是以 `cm` 起頭，代表 Computer Modern，以下將簡稱為 CM 字體。譬如，12 點之羅馬字體 Computer Modern Roman 檔名為 `cmr12.mf`；10 點之仿打字機字體 Computer Modern Typewriter 檔名為 `cmtt10.mf`。META-

FONT 軟體是以數學式精確地描繪每一個字母的外框，譬如，`cmr12.mf` 檔案內即包含約一百組程式，每一組程式描繪一個字母。

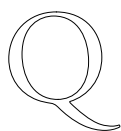
13.1.1 描邊字型與描點字型

排版時， $\text{T}_\text{E}\text{X}$ 僅需要每一個字母之寬度、高度等資訊，並不須知道字母實際的黑白點構成。但是，排版完成後，預覽與列印軟體就需要每一個字母之描點字型檔案，才能顯示或列印。描點字型檔可由描邊字型檔創造出來。METAFONT 描邊字型檔所產生之描點字型檔之延伸檔名為 `.pk`，譬如 `cmr12` 描點字型檔名為 `cmr12.pk`。在 $\text{MiK}\text{T}_\text{E}\text{X}$ 系統下，開啓 YAP 預覽/列印軟體時，顯示器有時會出現 “YAP is creating `cmr12.pk`...” 訊息，即表示正由 `cmr12.mf` 產生描點字型檔。 $\text{f}_\text{p}\text{T}_\text{E}\text{X}$ 之 `Windvi` 軟體也有類似的動作。

以描點字型預覽/列印，好處是速度快，缺點是不同印表機需不同的描點字型。舉例言之，如果你有一部 HP1100 印表機，精密度為 600dpi。為了加快列印速度，平常可能以 300dpi 列印初稿，等到最後一稿再以 600dpi 列印。如此一來，硬碟須同時儲存兩種精密度之描點字型檔。如果排版文稿最後印成書籍，以 1200dpi 列印，則電腦還須產生 1200dpi 描點字型檔。

雖然 PostScript 字體也是描邊字型檔，但可直接用於預覽與列印，不須先產生描點字型檔。具體言之，排版完成之後，先以 `DVIPS` 將 `DVI` 檔案轉換成 PostScript 格式，其內儲存文稿所使用的每一個英文字母或中文單字之描邊字型程式。以 `GSview` 預覽與列印時，軟體即時在記憶體內產生描點字型以供使用。因此，預覽與列印速度會稍慢，但好處是硬碟內不須儲存一大堆描點字型檔；而且，同一個 PostScript 檔案可在任何精密度之顯示器或印表機上顯示與列印。

描邊字型可能來自三種來源：一是 METAFONT 字型檔，二是 PostScript 字型檔，第三是 True Type 字型檔。以目前的發展而論，第二種的 PostScript 字型檔使用最廣，品質也可能最好。這三種字型檔都是所謂的描邊字型 (outline font)。描邊字型與描點字型的差別可以由本頁圖例說明。圖中上方是英文字母 Q 描邊字型的例子，它是由描繪出字母外框的線條所構成。當 YAP/`Windvi` 軟體要將排版結果呈現於顯示器上或列印時，它先使用工具程式由描邊字產生圖下方之描點字型。描點字型是由一個一個的小黑點構成，這就是我們在顯示器或紙上所看到的。



描邊字型



描點字型

1990年代初期, Microsoft 公司發展 Windows 作業系統時, 同時發展一套字型程式規格, 稱為 True Type。因為 Windows 系統之普及, True Type 字體也非常普及。目前中文 Windows 系統上使用之字體, 絕大部分都是 True Type 規格。Windows 文書軟體也是直接以描邊字型檔列印。譬如, Word 軟體列印文稿時, 也是先在記憶體內將 True Type 字體轉換成描點字型, 再進一步處理印出。但是, True Type 字體與 \TeX 之結合使用尚未十分成熟, 因此本章之介紹將以 PostScript 字體為主。

13.1.2 PostScript 格式之 CM 字體

許多繪圖軟體, 譬如, Corel Draw 與 Illustrator 都附有多種 PostScript 字型檔。Ghostscript 4.0版開始, 附有幾套各字體設計公司捐贈的 PostScript 字體。其中, 德國 URW++ 公司所捐贈的一整套字體, 品質相當不錯。因此, 有一些 PostScript 字體是可以免費使用的。但是, 如果你要排版的是數學文稿, 則問題是在於能用於排版數學式之 PostScript 字體很少。有些人排版一般文字時使用 PostScript 字體, 但數學式卻被迫使用 METAFONT 數學字體。但是, 每一種字體有其風格, 如果你使用 PostScript 之 Times 字體排版正文, 數學式卻使用 METAFONT 字體, 版面看來並不協調, 我們在稍後會以實際例子說明。

因為一般商用之 PostScript 字體無法提供足夠的數學符號, 有人即設法將 METAFONT 之數學符號字體轉換為 PostScript 格式。目前, CTAN 上至少有兩套免費使用之 PostScript 格式 CM 字體, 其中一套是由美國數學學會提供。利用此 PostScript 字型檔, 若以 CM 字體排版文稿, 預覽/列印時可使用描點或描邊字型檔。譬如, 以 YAP/Windvi 預覽或列印時, 使用描點字型。反之, 排版結果若先轉換為 PostScript 檔, 再以 GSview 預視/列印, 則是直接使用描邊字型。

本章的討論將會介紹一些有名的英文字體, 如 Garamond, Palatino 等。有意進一步探究的讀者, 請參考 [Bringhurst \(1996\)](#)。

13.2 中文 PostScript 字體

c\TeX 系統所使用的中文字體都是 PostScript 格式。一般的英文 PostScript 字體組, 如 Times Roman, 包含: 中體字、斜體字、粗體字、與粗斜體字共計 4 個檔案。 c\TeX 中文字體並無類似的字體組合; 但是, PostScript 字體可以作傾斜或水平

<code>\ctxfm</code>	—	明體正體字型
<code>\ctxfms</code>	—	明體斜體字型
<code>\ctxfbb</code>	—	粗黑體正體字型
<code>\ctxfbbe</code>	—	粗黑體狹長字型
<code>\ctxfbbes</code>	—	粗黑體狹長斜體字型

圖 13.1: 中文斜體與狹長字型

放大/縮小之變形。因此,由一種中文字體可以產生多種變化字體出來。譬如,選用中文粗黑體字之指令為 `\ctxfbb`;我們可由此產生三種變化字體:斜粗黑字體,指令為 `\ctxfbbs`;狹長粗黑字體,指令為 `\ctxfbbe`;以及兩種變形同時出現,指令為 `\ctxfbbes`。同理,明體斜體字之指令為 `\ctxfms`。中文字體經過變形處理之後,其效果請見圖 13.1 之例子。

中文狹長字體或斜體字的主要用途是在排版標題或者簡短的傳單海報上。如果使用狹長字體排版文稿段落,你會發現英文標點符號後之空白顯得太大;英文單字之間距也顯得太大。欲縮小英文字距與標點符號後之空白,可於段落之前使用下列指令:

```
\fontdimen2\font=0.9\fontdimen2\font
```

其中, `\fontdimen2` 為 \TeX 控制英文字距之指令。經過以上設定,字距將縮小為內定值的 90%。

13.2.1 重新定義中文變形字

中文斜體字或狹長字都是由正體字體變形而來。安裝 $\text{cw}\TeX$ 時,安裝程式會自動設定;其中,斜體字之傾斜度為 16.7 度,狹長字則是將原始字體水平縮小 91%。以下簡單說明如何定義中文字體之變形。

首先,我們說明字型度量檔的概念。排版時, \TeX 需要知道每一個字母或中文字之寬度與高度。每一個 \TeX 字體都有一個字型度量 (font metrics) 檔案,延伸檔名為 `.tfm`。譬如,英文 `cmr12` 字型度量檔名為 `cmr12.tfm`;中文字體 `m2` 之字型度量檔名為 `m2.tfm`。PostScript 字體也有其字型度量檔,以 `.afm` 為延伸檔名。譬如, `m2` 字型度量檔名為 `m2.afm`。 \TeX 之字型度量檔與 PostScript 字型度量檔內容並不同;不過,由後者可以創造出前者。

欲使用水平縮小(放大)或斜體字體,我們須創造 T_EX 系統之 .tfm 字型度量檔案。在 MiK_TE_X 或 fp_TE_X 中,中文描邊字型皆置於 c:\texmf\fonts\type1\cwtex 檔案夾內。以明體常用字為例,PostScript 描邊字型檔案名稱爲 m0.pfb 至 m51.pfb,共計 52 個;而字型度量檔之延伸檔名爲 .afm,置於 c:\texmf\fonts\afm\cwtex 檔案夾內。

T_EX 排版所需之 .tfm 字型度量檔案可以由 PostScript 之 .afm 字型度量檔案轉換出來,轉換方法是利用 DVIPS 所附之工具程式: afm2tfm。以 m2 字型檔為例,欲產生斜體字 ms2 之 ms2.tfm 字型度量檔,請先進入存放 PostScript 字型度量檔之檔案夾內,執行:

```
c:\texmf\fonts\afm\cwtex>afm2tfm m2 -s 0.167 ms2
```

將由 m2.afm 產生 ms2.tfm,其中,-s 0.167 設定傾斜度爲 0.167。此一角度是普遍使用之值,但我們可以任意選擇,甚至負值也可以使用。

依同法產生全部 52 個字型度量檔案後,將它們全部移入 c:\texmf\fonts\tfm\cwtex 檔案夾內。檔案移入之後,須更新檔案資料庫;MiK_TE_X 使用者請於 DOS 模式內執行:

```
c:\>initexmf -u
```

若安裝 fp_TE_X,請執行:

```
c:\>mktexlsr
```

13.2.2 字型對應檔

以上設定已足以讓 T_EX 作排版工作,但要順利執行 DVIPS,還須進一步設定。在 c:\texmf\dvips\cwtex 檔案夾內可找到 cwtex.map 字型對應檔,以文字編輯軟體叫出,其內容如下:

```
m0 CWTEX-M0 <cwtex.enc <m0.pfb
m1 CWTEX-M1 <cwtex.enc <m1.pfb
m2 CWTEX-M2 <cwtex.enc <m2.pfb
...
```

第一行指令說明 m0 字體之全名爲 CWTEX-M0,每一個單字在檔案內之排列是依照

`cwtext.enc` 檔案所指定之順序; 列印時應自硬碟中取用 `m0.pfb` 字型檔。同理, 第 2-3 行分別定義 `m1` 與 `m2` 等字體。

實際列印時, 假設 DVIPS 發現文稿內使用了 `m1` 字體中第 115 個單字, 它即自硬碟內尋找 `m1.pfb` 字型檔案, 依據 `cwtext.enc` 所排定之順序, 找出第 115 個單字之描邊程式, 將之轉換為一般之 PostScript 格式; 下一階段再由 GSview 印出。

為了方便區分, `cwtext.map` 檔案僅包含原形字體之對應指令; 變形字之對應設定則置於 `cwtext1.map` 檔案內。明體斜體字形事實上是由正體字轉換而來, 因此 `cwtext1.map` 檔案也須設定 27 行定義。再以前三個字體為例, 指令如下:

```
ms0 CWTEX-M0 ".167 SlantFont" <cwtext.enc <m0.pfb
ms1 CWTEX-M1 ".167 SlantFont" <cwtext.enc <m1.pfb
ms2 CWTEX-M2 ".167 SlantFont" <cwtext.enc <m2.pfb
...
```

第一行指令之意義為: 斜體字 `ms0` 之全名為 `CWTEX-M0`, 列印時應自硬碟中取用 `m0.pfb` 字型檔, 但字體應傾斜 0.167 度。文稿中若使用 12pt 之斜明體字 `ms2`, DVIPS 程式將取用 `m2.pfb` 字型檔, 並作指定之變形。

如圖 13.1 所示, 除了斜體字之外, 我們也可以將字體水平放大或縮小。若粗黑體字要水平縮小 91%, 首先須製造 `.tfm` 字型度量檔。以 `bb3` 字體為例, 執行之指令為:

```
c:\texmf\fonts\afm\cwtext>afm2tfm bb3 -e 0.91 bbe3
```

其中 `-e 0.91` 表示字體將水平縮小 91%; 此比率數字可以任意選定。若數值小於 1, 變形字體將較原字體為瘦; 若參數值大於 1, 變形字體將較胖, 其次, `cwtext1.map` 檔案內亦須添加 27 行定義, 指令如下:

```
bbe0 CWTEX-M0 "0.91 ExtendFont" <cwtext.enc <bb0.pfb
bbe1 CWTEX-M1 "0.91 ExtendFont" <cwtext.enc <bb1.pfb
bbe2 CWTEX-M2 "0.91 ExtendFont" <cwtext.enc <bb2.pfb
...
```

其中, `"0.91 ExtendFont"` 表示字體應水平縮小 91%。

經過以上設定, `\ctxfms` 可用以排出明斜體字, `\ctxfbbe` 則排出水平縮小 91% 之粗黑體。若是水平縮小與傾斜同時作用, 則須一起加入上兩項設定, 請

參見 `cwtex1.map` 檔案。該檔案置於 `c:\texmf\dvips\cwtex` 檔案夾內。有關於 `afm2tfm` 之使用方法, 請見 13.5.2 節進一步的說明。

13.3 英文 PostScript 字體

大部分的 PostScript 字體都是商業軟體, 但也有少部分可以自網路下載, 免費使用。Adobe 公司的 Times New Roman PS 字體是一商業軟體, 但該公司的 Acrobat Reader 可免費下載, 其中即附有此套字體。因此, 電腦安裝 Acrobat Reader 時, 即同時安裝此一字體。Times New Roman PS 字體族計有中體、斜體、粗體、斜粗體等 4 種, 分為 4 個字型檔案。取得了 4 個 PostScript 字體之後, 我們如何使用它們來排版呢? 這個問題看來簡單, 其實其中牽涉許多問題。

首先, \TeX 排版時需讀取字型度量檔 `.tfm` 之字體資訊, 但 PostScript 字體僅提供 `.afm` 字型度量檔。但此一問題不難解決。上一節曾介紹 `afm2tfm` 程式, 可用來將 `.afm` 檔案轉換為 `.tfm` 檔案。其次, \TeX 所使用之 METAFONT 字體, 一個字型檔含 128 個字母; 但一般 PostScript 字型檔約含有 220 個字母。除了字母個數不同之外, 字母排列順序也不完全相同。換言之, 兩種字型檔字元排序 (encoding) 方式不同。如果直接取用 PostScript 字型檔內之字母, 有可能把英磅符號 £ 排版成貨幣符號 \$。

解決以上問題的辦法是透過巨集套件。例如, 若硬碟中已安裝 Times 描邊字型檔案, 則利用 \TeX 所提供的 `times` 巨集套件, 即可使用此一字體。輸入文稿時, 在全文設定區加入下列一行指令:

```
\usepackage{times}
```

文稿之正文即全部改用 Times 字體排版。以上指令引入 `times` 巨集套件, 它是 PSNFSS 的一部分, 作者為 Sebastian Rahtz。

PostScript 字體數以千計, 但常用的並不多。而且, 不同字體公司可能提供同一檔名之字體, 但卻各有獨特的設計。譬如, 大部分字體公司都提供 Garamond 字體, 但每一家各有其獨特設計的風格。此外, 新字體每年不斷出現。因此, 要對每一種字體都提供巨集套件顯然有困難。PSNFSS 所提供的巨集套件主要是針對使用率較高的字體。譬如, 一般 PostScript 印表機內附有 Adobe 公司的 35 種字體, 包括: Times Roman, Palatino 等, 這可能是最普遍使用的 PostScript 字體。

PSNFSS 巨集套件主要功能即使用這些字體排版。

顧名思義, PSNFSS 是架構在新版 \TeX 的 NFSS 字體選用法之上。每一種字體依其特徵可以作底下的分類: 字體族 (font family), 字體型 (font shape), 字體序列 (font series) 等。譬如, 在 Times 的字體族之下, 我們可以選用中體字序列, 也可以用粗體字序列; 可以選用正體字形, 也可以用意大利斜體字形。萬一我們所購置之 PostScript 字型檔不足, 則當文稿中選用硬碟中所沒有的字形時, PSNFSS 會自動選用替代字形。

上文提及, 有些公司捐贈了幾套 PostScript 字體免費提供使用。其中, URW++ 公司所捐贈之字體即涵蓋了 Adobe 的 35 種標準字體。此外, Adobe 公司所捐贈之 Utopia 字體及 Bitstream 公司所捐贈之 Charter 字體, 品質也相當好。

13.3.1 txfonts 巨集套件

軟體公司設計出售各種英文字體, 但其中絕大部分僅含普通英文字母, 排版數學式字所需之符號並不包含在內。Young Ryu 設計了兩組排版數學文稿所需之字型, 第一套稱為 `txfonts`, 第二套稱為 `pxfonts`。前者的字體設計風格接近 Times Roman 字體, 後者接近 Hermann Zapf 所設計的 Palatino 字體。使用 `txfonts` 巨集套件很簡單, 僅須在全文設定區加入下列指令即可:

```
\usepackage{txfonts}
```

若文稿內並無數式, 以上指令仍將正文改以 Times Roman 字體排版。

Ryu 所提供的 `pxfonts` 巨集套件是以 Palatino 字體為基礎的數學字體。除此之外, PSNFSS 內的 `mathpazo` 也是以 Palatino 字體為基礎的數學字體, 作者為 Diego Puga。圖 13.2 比較四種正文與數式字體。最上一列是以 Knuth 所設計的 METAFONT 字體排版的結果。第 2 列是 `txfonts` 所排版之結果; 除了筆劃粗細不同之外, 某些字型設計也不一樣。第 3 列是 `mathpazo` 所排版之結果; 正文字體與數學符號都是另一種風格。

最底下一列之排版, 正文字體使用 Adobe Garamond, 數學符號則使用 Math-time 字體。這兩套字體都是商業軟體, 後者是由 Y&Y 公司發展, 相關資訊請見網站:

<http://www.yandy.com>

Computer Modern math fonts:

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx$$

$$a_0 + \left(\sum_{t=0}^{\infty} \frac{y_t}{1+r} - \sum_{t=0}^{\infty} \frac{c_t}{1+r} \right) = 0$$

ABCDEFGHIJKLMN 1234567890 mnopqrstuvwxyz

txfonts package:

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx$$

$$a_0 + \left(\sum_{t=0}^{\infty} \frac{y_t}{1+r} - \sum_{t=0}^{\infty} \frac{c_t}{1+r} \right) = 0$$

ABCDEFGHIJKLMN 1234567890 mnopqrstuvwxyz

mathpazo package:

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx$$

$$a_0 + \left(\sum_{t=0}^{\infty} \frac{y_t}{1+r} - \sum_{t=0}^{\infty} \frac{c_t}{1+r} \right) = 0$$

ABCDEFGHIJKLMN 1234567890 mnopqrstuvwxyz

Adobe Garamond plus Mathtime math fonts:

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx$$

$$a_0 + \left(\sum_{t=0}^{\infty} \frac{y_t}{1+r} - \sum_{t=0}^{\infty} \frac{c_t}{1+r} \right) = 0$$

ABCDEFGHIJKLMN 1234567890 mnopqrstuvwxyz

圖 13.2: 四種正文與數式字體之比較

13.4 創造英文 PostScript 字體巨集套件

字體是排版品質的關鍵。即使是中文稿，同一中文字體搭配不同的英數字體即可產生完全不同的效果。大部分的英文繪圖軟體都附有 PostScript 字體，譬如，英文版 Corel Draw 即附有數百種字體，Adobe Illustrator 軟體亦附有多種字體。

取得字型檔後，我們須透過巨集套件才能使用這些字體。Ghostscript 附有德國 URW++ 公司所捐贈的數十套字型檔案，下一小節將以其中的 Palatino 及 Helvetica 兩套字體為例，說明創造 PostScript 巨集套件的方法。

13.4.1 fontinst 巨集套件

創造 PostScript 字體巨集套件最好的方法是使用 Alan Jeffrey 與 Rowland McDonnell 所寫的 fontinst 巨集套件。一套英文 PostScript 字體通常包含四個 .pfb 字型檔案，分別對應正體、斜體、粗體與粗斜體字。另外，我們還需要字型度量檔案 .afm。如果軟體公司並未提供字型度量檔，我們可以從 CTAN 網站上搜尋。萬一連網站上都找不到，我們還可使用 Ghostscript 所提供之工具程式 pf2afm；或者利用字體設計軟體如 Fontographer 產生。

若電腦內已安裝 Ghostscript 7.04 版，則 URW++ 所捐贈之字型檔案應置於 c:\gs\fonts 檔案夾內。底下先說明如何創造 Palatino 字體之巨集套件。此一字體族含有 4 個描邊字型檔，檔名如下：

```
p052003l.pfb (正體字)
p052023l.pfb (斜體字)
p052004l.pfb (粗體字)
p052024l.pfb (粗斜體字)
```

Ghostscript 7.04 版已附有字型度量檔案，但其他繪圖軟體所附字型不見得含有字型度量檔；故底下仍簡單說明如何自行產生字型度量檔。

硬碟內先創造 c:\texmf\fonts\type1\palatino 檔案夾，將以上四個字型檔案移入其中。以文字編輯軟體開啓 c:\gs\gs7.04\lib 檔案夾內之 pf2afm.bat 檔案，將最後一行指令更改如下：

```
c:\gs\gs7.04\bin>gswin32c -q -sDEVICE=nullpage -- pf2afm.ps %1
```

為了方便起見，將此批次檔與上述四個字型檔案都複製於 `c:\xtemp` 檔案夾內執行下列指令：

```
c:\xtemp>pf2afm p052003l
```

硬碟內將產生 `p052003l.afm`。依同法，產生其他三個字型度量檔案。

取得字型度量檔案後，下一步驟是須更改檔名。為了避免混淆，數以千計的 PostScript 字體必須有一套命名方法。目前通用的重新命名方法是由 Karl Berry 所設計者。在此命名之下，Palatino 字體名字簡化為 `pl` 兩字；Helvetica 字體則簡化為 `hv`。許多字體設計公司各有其 Palatino 字體，因此字體名稱上還須標示公司名稱以資區別。德國 URW++ 軟體公司是以字母 `u` 代表，因此以上兩種字體將分別改名為 `upl` 與 `uhv`。有關於各種字體之重新命名方法，請見 `\texmf\fontname` 檔案夾內之說明檔。

依以上原則，Palatino 4 個字型度量檔案應分別改名如下：

```
p052003l.afm ==> uplr8a.afm
p052023l.afm ==> uplri8a.afm
p052004l.afm ==> uplb8a.afm
p052024l.afm ==> uplbi8a.afm
```

新檔名末端之 `8a` 是用於標示字元排序 (encoding) 的方式。檔名中之 `r` 為 `regular` 簡寫，代表正體；`i` 為 `italic` 簡寫，代表斜體，`b` 為 `bold` 簡寫，代表粗體；`bi` 則代表粗斜體。

接下來，我們須利用 `fontinst` 巨集套件進一步處理。進入 DOS 模式，執行下列指令：

```
c:\xtemp>tex fontinst.sty
```

執行以上指令之後，程式將讀取檔案夾內之 4 個字型度量檔案，並由之創造 TeX 排版時所需之各種檔案。當顯示器上出現 * 號，應鍵入：

```
\latinfamily{upl}{-}\bye
```

程式執行之後，硬碟中將出現三組檔案，延伸檔名分別為 `.pl`，`.vpl`，與 `.fd`；前兩組檔案須進一步轉換。

欲處理這兩組字型度量檔案需要 `pltotf.exe` 與 `vptovf.exe` 兩個程式。如果你使用 `MiKTeX` 我們建議改用 `emTeX` 所提供者，原因是執行較方便。(如果你使用 `fpTeX`，可直接執行。) 光碟 `c:\support` 檔案夾內有 `hlatex.zip`，其內存放有這兩個程式，請將檔案複製至 `c:\xtemp` 內，執行下列兩道指令：

```
c:\xtemp>for %f in (*.pl) do pltotf %f
c:\xtemp>for %f in (*.vpl) do vptovf %f
```

以上指令將創造出十幾個 `.tfm` 與 `.vf` 檔案，請將所有的 `.tfm` 檔案複製於 `c:\texmf\fonts\tfm\palatino` 檔案夾內。請注意，`... \tfm\palatino` 次檔案夾原本並不存在，必須先創造之。同樣的，先創造 `... \vf\palatino` 次檔案夾，再將所有的 `.vf` 檔案複製於其內。

在 `\xtemp` 檔案夾內另可找到 4 個 `.fd` 檔案，其中 `OT1upl.fd`，`T1upl.fd` 與 `TS1upd.fd` 三個字體驅動檔案，以文字編輯軟體開啓 `OT1upl.fd`，其內容略加整理之後有下列之指令：

```
...
\DeclareFontFamily{OT1}{upl}{}
\DeclareFontShape{OT1}{upl}{b}{n}{ <-> uplr7t}{}
\DeclareFontShape{OT1}{upl}{b}{sc}{ <-> uplrc7t}{}
...
```

每一行指令設定某種字體的使用方法。譬如，第 2 行指令即指定以 `uplr7t` 字型檔排版 `Palatino` 粗體字，並使用 `OT1` 之字元排序方式。每一種字體有其特定之設計尺寸。若一篇文稿中混用幾種字體，即使同樣使用 12 點字體排版，不同字體之間可能會大小不一。因此，字體大小須略作調整。

經過幾次嘗試，我們發現 `upl` 須略縮小才能與中文字體搭配。因此，`OT1upl.fd` 檔案內容須修改如下：

```
...
\DeclareFontFamily{OT1}{upl}{}
\DeclareFontShape{OT1}{upl}{b}{n}{ <-> [.94] uplb7t}{}
\DeclareFontShape{OT1}{upl}{b}{sc}{ <-> [.94] uplbc7t}{}
...
```

各行內所加上之 `[.94]` 之作用是將指定字體縮小為原尺寸的 94%。修改完畢之後，先創造 `c:\texmf\tex\latex\palatino` 檔案夾，再將 4 個 `.fd` 檔案移入其中。

Tlupl.fd 與 TS1upd.fd 請以同樣方法處理。Palatino 字體是屬於 serif 字體，我們另選取一 sans serif 字體搭配使用。

在 URW++ 所提供的字體中，Helvetica 是一個不錯的選擇。依照 Karl Berry 所設計之命名規則，此字型檔名應以 `uhv` 開頭，因此 4 個字型度量檔應重新命名如下：

```
n019003l.afm ==> uhvr8a.afm
n019023l.afm ==> uhvri8a.afm
n019004l.afm ==> uhvb8a.afm
n019024l.afm ==> uhvbi8a.afm
```

依照處理 Palatino 字型檔的方法，對於 Helvetica 字型檔同樣處理一遍。如此，硬碟內已存有使用兩種 PostScript 字體所需之檔案。最後，以文字編輯軟體寫出 `uplhv.sty` 檔案，內容如下：

```
%% This is file 'uplhv.sty',
\renewcommand{\rmdefault}{upl}
\renewcommand{\sfdefault}{uhv}
\endinput
%% End of file 'uplhv.sty'.
```

第 1 行指令設定 `upl` 字體作為正體字，第 2 行指令設定 `uhv` 字體作為 sans serif 字體。此檔案應移入 `c:\texmf\tex\latex\palatino` 檔案夾內。

以上所製作的檔案可供 \TeX 排版之用，但為了讓預覽/列印軟體能找到所需的字型檔案，尚須製作一字型對應檔 (font mapping file)。表 13.1 為字體對應檔案之內容，檔案取名為 `uplhv.map`。請特別注意，因為版面寬度容納不下全部文字，我們將其中字元串 `TeXBase1Encoding ReEncodeFont` 暫以字母 `x` 替代。實際製作檔案時，字母 `x` 應替代回原始字串。

字型對應檔案內每一行指令內含有 5 個段落，其中，第 1 個段落為 \TeX 排版時所認定之字體名稱。以第 1 行為例，檔名為 `uplr8r`，代表 Palatino 正體字。第 2 個段落為 `URWPalladioL-Roma`，此字體名取自 `.pfb` 字型檔內之 `/FontName` 指令行，這是 PostScript 字體之正式名稱。第 3 與第 4 段落設定字型檔內各字母之排序方法，最後一個段落為硬碟內描邊字型檔之名稱。由表 13.1 可以看出，每一字體族須作 6 行設定。前 4 行分別為正體字、斜體字、粗體字，與粗斜體字；第 5 行為數學斜體字，第 6 行為數學粗斜體字。

表 13.1: uplhv 巨集套件之字型對應檔案

uplr8r	URWPalladioL-Roma "x" <8r.enc <p052003l.pfb
uplri8r	URWPalladioL-Ital "x" <8r.enc <p052023l.pfb
uplb8r	URWPalladioL-Bold "x" <8r.enc <p052004l.pfb
uplbi8r	URWPalladioL-BoldItal "x" <8r.enc <p052024l.pfb
uplro8r	URWPalladioL-Roma ".167 SlantFont x" <8r.enc <p052003l.pfb
uplbo8r	URWPalladioL-Bold ".167 SlantFont x" <8r.enc <p052024l.pfb
uhvr8r	NimbusSanL-Regu "x" <8r.enc <n019003l.pfb
uhvri8r	NimbusSanL-ReguItal "x" <8r.enc <n019023l.pfb
uhvb8r	NimbusSanL-Bold "x" <8r.enc <n019004l.pfb
uhvbi8r	NimbusSanL-BoldItal "x" <8r.enc <n019024l.pfb
uhvro8r	NimbusSanL-Regu ".167 SlantFont x" <8r.enc <n019003l.pfb
uhvbo8r	NimbusSanL-Bold ".167 SlantFont x" <8r.enc <n019024l.pfb

說明: 表中之 x 應以 TeXBase1Encoding ReEncodeFont 字串替代。

檔案 uplhv.map 製作完成之後, 應創造 \texmf\dvips\uplhv 檔案夾, 將字體對應檔移入其內。最後, 以文字編輯軟體開啓 \texmf\dvips\config 檔案夾內之 config.ps 檔案, 其中可找到下列設定行:

```
p +cwtex.map
p +utopia.map
...
```

請加入:

```
p +uplhv.map
```

所有設定即大功告成。最後, 我們須更新檔案資料庫。MiKTeX 使用者請執行:

```
c:\xtemp>initexmf -u
```

fpTeX 使用者請執行:

```
c:\xtemp>mktexlsr
```

最後, 以上的處理過程會在 c:\xtemp 內留下許多輔助檔案, 為節省空間可將之全數刪除。

字體對應檔之運作原理可簡單說明如下: 文稿若使用 uplhv 巨集套件排版,

正文將取用 Palatino 正體字。文稿排版完成轉換為 PostScript 格式時, DVIPS 程式發現文稿內使用 `uplr8r` 正體字, 開始在硬碟內尋找描邊字型檔。DVIPS 首先檢查 `config.ps` 檔案內每一個字體對應檔案, 當檢查到 `uplhv.map` 時, 發現檔案內含有 `uplr8r` 字體之對應設定, 即循此自硬碟內讀取 `p052003l.pfb` 描邊字型檔, 並進行轉換工作。

13.4.2 選用字體

巨集套件創造出來後, 排版時只須在全文設定區引用巨集指令:

```
\usepackage{uplhv}
```

TEX 即以 Palatino 作為正文字體, 以 Helvetica 作為 sans serif 字體。以上指令會改變整篇文稿之字體。若只要改變文稿中某一段落之字體, 可使用下列指令:

```
{\fontfamily{upl}\selectfont text}
```

大括號內之 *text* 將改用 Palatino 字體。以上之 `\fontfamily` 指令用以選用字體族。除此之外, 我們也可以改變字體系列, 指令為 `\fontseries`。同理, `\fontshape` (字形) 指令用以選擇正體、斜體; `\fontsize` 指令則用以選擇字體大小與行距。

字形指令 `\fontshape` 可使用 4 個選項: `n`, `it`, `sl`, 與 `sc`; 其中, `n` 為正常形 (normal), `it` 為斜體形, `sl` 代表數學斜體字, `sc` 代表 small capital 字體。字體系列指令 `\fontseries` 同時標示重量 (weight) 與寬度 (width) 兩項特徵。其中, 重量是指筆畫粗細, 譬如, 粗體字為 `b`, 標準寬度為 `m`。字體寬度由極小之 `uc` (ultra-condensed) 到極大之 `ux` (ultraexpanded); 標準或正常寬度為 `m`。如果選用標準寬度, `m` 指令可省略, 譬如, 標準寬度之中體字指令為 `\fontseries{m}`; 又如, 標準寬度之粗體字指令為 `\fontseries{b}`; 相對的, 狹長 (condensed) 中體字指令為: `\fontseries{mc}`, 其中, `m` 代表中體字, `c` 代表 condensed。

根據以上說明, 欲選用 Charter 粗黑體, 指令為:

```
\fontfamily{bch}\fontseries{b}\selectfont
```

其中, `bch` 後兩個字母代表 Charter, 字母 `b` 則代表設計字體之 Bitstream 公司。如果在 `\selectfont` 指令之前又加入 `\fontsize{15}{20pt}`, 則字體大小將變

為 15pt, 行距為 20pt。若再加入 `\fontshape{it}` 指令, 即指示斜體字。再舉一個例子, 若輸入以下指令:

```
{\fontfamily{bch}\fontseries{b}\fontshape{sc}\selectfont Dvips}
```

即可排出 Charter 字體族, 粗體系列, 標準寬度, 及 small capital 字型。

上一節曾介紹 charter 巨集套件, 其功能是設定羅馬字體選用 Charter 字體; 但 sans serif 仍使用 \TeX 原始字體。如果要把 charter 巨集套件內之 sans serif 字體也改用 Helvetica 字體, 只要將 charter.sty 檔案依前面 uplhw.sty 作類似修改即可。

13.4.3 使用 True Type 字體

每一套 Windows 作業系統是都附有一些英文 True Type 字體, 例如 Times New Roman, Arial 等。目前已有有人嘗試要讓 \TeX 使用 True Type 字體排版, 但迄今為止尚不十分成熟。

目前的發展中, pdf \TeX 的作法是直接取用 True Type 字型檔。另外一種作法則是先將 True Type 字型檔轉換為 PostScript 字型檔, 再利用 fontinst 產生所需之巨集套件。網路上有幾套免費下載程式, 可將 True Type 字型檔轉換成 PostScript 字型檔, 其中之一名為 `tt2pfb`; MiK \TeX 與 fp \TeX 系統內都含有此一軟體。有興趣嘗試者, 可至 Free Type 網站取的進一步資訊:

<http://www.freetype.org>

13.5 DVIPS 與 psutils 工具程式

欲使用 PostScript 圖形或 PostScript 字型, 我們必須將排版結果轉換為 PostScript 格式。目前使用最廣的轉換驅動程式可能是 DVIPS, 此為 Tom Rokicki 所創作。底下兩小節分別介紹 DVIPS 與附屬工具程式 `afm2tfm` 之功能。

13.5.1 DVIPS 程式

DVIPS 程式的主要功能是將 \TeX 排版結果轉換為 PostScript 格式。若使用 cw \TeX 之設定, 在 WinEdt 視窗內按下功能鍵 [F11], 即執行 DVIPS 將排版文稿全文轉

換為 PostScript 格式。不過，有時候我們僅換某幾頁，或者須轉換為特別格式，則執行程式時須加入選項。

DVIPS 程式之指令格式如下：

```
c:\>dvips - option file.dvi
```

其中，file.dvi 為 T_EX 排版結果，延伸檔名 .dvi 可省略；-option 為程式選項，底下列出 DVIPS 較常用之選項。

- p 列印範圍之首頁。譬如，選項 -p 3 即選擇自第 3 頁開始列印。
- l 列印範圍之末頁，譬如，dvips -l 8 即選擇列印至第 8 頁。綜合以上兩選項，-p 3 -l 8 即指示列印的 3-8 頁。
- pp 另一種選擇列印範圍的方法。譬如，-pp 3,5,7:10 選擇列印第 3, 5 兩頁，及 7-10 頁。單獨之頁碼以逗號分開，連續之頁碼以冒號區隔首末頁。
- n 設定總列印頁數。譬如，-p 3 -n 20 表示自第 3 頁開始，共列印 20 頁，至第 22 頁為止。
- o 設定列印檔名。不加設定時，若文稿原名為 file.dvi，列印結果為 file.ps。
- O 調整列印在紙面的位置。此選項須設定 X,Y 兩項座標數字。例如，若列印區域要自原先設定位置往右移動 1 公分，並且下移 0.5 公分，應加入之選項為 -O 1cm, 0.5cm。(我們也可以直接在 GSview 軟體內調整列印位置。)
- E 列印為 Encapsulated PostScript 格式。加上此選項後，所產生之檔案內將包含有 %BoundingBox 指令。

除了以上所列之外，DVIPS 還提供許多選項，請參考其說明檔。

13.5.2 afm2tfm 程式

Tom Rokicki 另外寫了一個工具程式 afm2tfm，其功能是将 PostScript 字體之 .afm 字型度量檔案轉換為 T_EX 排版時所需之 .tfm 字型度量檔。前面介紹中文變形字體時曾說明其應用方法。此一程式之指令格式如下：

```
c:\>afm2tfm inputfile - option outputfile
```

`infile` 是指 PostScript 字型度量檔, 例如, `ptmr8r.afm` 即為 Times 字型度量檔名。輸入時, 附屬檔名可省略; `outfile` 為轉換結果之檔名。若省略輸出檔名, 程式將自動取名為 `ptmr8r.tfm`。

常用之選項有下列兩個:

- e 字體水平放大或縮小。例如 `-e 0.91` 設定將字體內每一個字之寬度壓縮為 91%。
- s 字體傾斜。例如, `-s 0.167` 設定將字體向右傾斜 16.7 度。若數字為負值, 字體將向左傾斜。

其他選項尚可控制字元排序 (encoding) 方法, 請見說明檔。

13.5.3 psutils 工具程式

排版是以一頁為基本單位, 但列印時有時候需要把版面順序重新調整。譬如說, 為方便列印於全開的紙面上, 排版者須把 9 頁合併於一個大版面中; 台灣排版業者稱此為「落大盤」。Angus Duggar 寫了一套 PostScript 工具程式, 主要功能是重新組合版面, 本節簡單介紹此套程式之功能。

工具程式 `psutils` 包含好幾個程式, 一般人最常用的程式是 `psnup`, 其功能是集合兩頁 (或多頁) 列印於一頁版面上。另一個較常用的程式是 `psbook`, 其功能也是調整版面順序; 新順序方便作書本裝訂。

● 數頁合併於一頁: `psnup` 程式

`psnup` 程式的應用是將兩頁版面列印於一頁上, 以節省用紙, 指令如下:

```
c:\xtemp>psnup -2 infile.ps outfile.ps
```

其中, `-2` 表示每兩頁併為一頁, `infile.ps` 為原檔名, `outfile.ps` 為新檔案名稱。若要 4 頁合併於一頁中, 選項 `-2` 應改為 `-4`。

此程式可加入許多選項以控制列印結果, 擇其要者簡介如下:

```
psnup -n -ppaper -Ppaper -sscale -mmargin infile outfile
```

第一選項 `n` 項代入數字, 表示要將 `n` 頁併入於一頁中。第 2 選項 `-p` 供選用輸出版面之紙張大小, 可填入 `a4`, `letter`, `b5` 等等。第 3 選項 `-P` 之功能與第 2 選項相

同,但指的是原文稿之紙張尺寸。若不加入紙張尺寸選項,程式假設皆為 A4 紙張。第 4 選項 `-s` 設定原版面縮小之比例。例如, `-s0.75` 指示版面縮小為原尺寸之 75%。最後一個選項 `-m` 設定輸出版面四周所留出之空白。例如, `-m1cm` 設定紙面四周各留下一公分空白。

● 調整版面順序: `psbook` 程式

若某一篇文稿共計 8 頁,列印時將依 1-8 頁之順序。現假設我們利用上面介紹之 `psnup` 程式每兩頁合併為一頁,則新的第一頁內有 1-2 之頁碼;第 2 頁上有 3-4 之頁碼等。列印之後,將此 4 頁疊在一起,從中間對折可變成一本小冊子。但我們會發現此小冊子之頁碼順序不正確。譬如,小冊子封面上出現的是第 2 頁而不是第 1 頁。

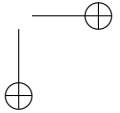
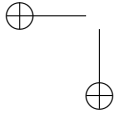
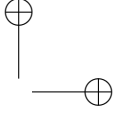
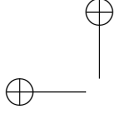
`psbook` 程式之功能即是將頁碼順序重新調整,使小冊子之頁碼順序得以正確出現。舉例言之,若文稿原檔名為 `file.ctx`,排版後得 `file.ps`。先執行 `psbook` 調整順序:

```
c:\xtemp>psbook file.ps file1.ps
```

再執行 `psnup` 將 8 頁合併為 4 頁:

```
c:\xtemp>psnup -2 file1.ps newfile1.ps
```

將 `newfile1.ps` 列印出來,4 頁疊在一起,中間對折,即可得到一正確頁碼順序之小冊子。



14 巨集指令

巨集指令 (macros) 就是把用來完成特定工作的一連串指令集合在一起, 並給予一個名稱。文稿內若使用此巨集指令, 排版時 \LaTeX 將依序執行其中的指令。定義巨集指令的用途之一是節省輸入長串指令的時間, 避免輸入錯誤。此外, 排版長篇文稿時, 前後的版面格式必須一致; 譬如, 各章節標題須使用同一字體。欲解決此一問題, 最好是利用巨集指令。

事實上, \LaTeX 本身就是由 \TeX 原始指令所定義出來的一套龐大的巨集指令。本書所介紹的各種巨集套件也是各種巨集指令組合而成。欲深入了解巨集指令的概念, 請參考 [Knuth \(1990\)](#) 或 [Kopka and Daly \(1995\)](#)。本章僅介紹最簡單的巨集指令的概念與用法, 最後一節說明排版本書所使用之巨集指令, 以供參考。

14.1 定義巨集指令

巨集指令的功能之一是用以節約輸入長串指令之時間。舉例言之, 若文稿內經常使用 `\medskip` 指令, 我們可以在全文設定區作加入下列定義:

```
\newcommand{\ms}{\medskip}
```

\LaTeX 使用 `\newcommand` 以定義巨集指令。第一個大括號內為巨集指令名稱。取名時, 請注意勿與現有之巨集同名。巨集指令名稱以反斜線起頭, 但名字只能使用大小寫英文字母, 不能使用數字。

因為 \LaTeX 本身是由 \TeX 指令組合而成, 因此, 我們也可以直接使用 \TeX 指令定義巨集指令。若使用 \TeX 指令, 上述之定義可以簡化為:

```
\def\ms{\medskip}
```

\TeX 之定義指令為 `\def`, 巨集指令名稱則直接附加於其後, 前後不須加上大括號。經過以上定義, `\ms` 指令即等於是 `\medskip`。

另外一個例子，論文裡經常須列舉他人之研究文獻。以下之巨集指令 `\laref` 設定一個排版參考文獻的格式：

```
\newcommand{\laref}{\par\noindent\hangindent=\parindent}
```

此一巨集指令之意義如下：定義一開頭以 `\par` 指令結束前一段落；接下來以 `\noindent` 指令設定下一行開頭不內縮；最後的 `\hangindent` 指令設定每一項文獻的第2行開始內縮 `\parindent` 距離。

經過以上定義之後，排版時可將 `\laref` 指令置於每一項文獻前端，每一文獻的第一行將由版面左緣開始編排，第二行以後每一行都會內縮一點距離。如果兩項文獻之間距還要加大一些，可在以上定義中 `\par` 指令之後加入 `\smallskip`。

再舉一個巨集指令之應用例子，本書經常使用的 `cwTeX` 標誌符號是由下列巨集指令所定義：

```
\newcommand{cw}{\tt cw}\TeX}
```

定義一開始首先選用打字機字體排版 `cw` 兩個字母，其後再加上 `\TeX` 指令。如此，文稿內輸入 `\cw{}` 即可排出 `cwTeX`。

與 `\newcommand` 指令類似的是 `\renewcommand`。前者用於定義新的巨集指令，後者則用於修改原已存在之巨集指令的內容。譬如，`ETEX` 排版摘要時，`ETEX` 會自動加上標題：**Abstract**，這是由 `\abstractname` 所定義。排版中文稿時，我們可使用下列指令重新定義為中文標題：

```
\renewcommand{\abstractname}{ 摘要}
```

巨集指令可由使用者自行加入變數。譬如，使用 `ETEX` 之迷你指令環境時，我們須設定迷你版面的寬度。若經常使用此項指令，且版面寬度大小不一，我們可以定義下列兩道巨集指令：

```
\newcommand{\bmp}[1]{\begin{minipage}{#1\textwidth}}
\newcommand{\emp}{\end{minipage}}
```

第二道巨集指令很簡單，以 `\emp` 指令替代較長的 `\end{minipage}`。第一道巨集指令 `\bmp` 定義迷你版面之開端，其中之 `[1]` 設定表示使用者須自行加入一數字。執行時，此一數字即代入指令後面的 `#1`。定義巨集指令之後，若輸入 `\bmp{0.4}`，

其作用與 `\begin{minipage}{0.4\textwidth}` 完全相同。

巨集指令可添加之變數不限於一項。若有兩項變數，則巨集指令名稱之後變成 [2]，定義內容分別以 #1 與 #2 代表這兩項變數。第 9 章曾說明排版迴歸方程式之巨集指令 `\tb` (頁 170)。原指令是以 `\def` 定義；若改用 \TeX 之 `\newcommand` 指令定義，應為：

```
\newcommand{\tb}[2]{\mathop{\#1\phantom{\sum}}%
\limits_{\displaystyle #2}}
```

\TeX 可以定義更複雜的巨集指令。譬如，巨集指令可以加入選項；或者在符合特定條件下才執行某項動作。詳細說明請見 [Kopka and Daly \(1995\)](#)，頁 189–208，與 [Goossens et al. \(1994\)](#)。

14.1.1 設定字級之巨集指令

排版文稿時經常須變更字級與行距。 \TeX 提供相對字級指令，如 `\small`，`\large` 等等。但我們也可以直接選用特定字級。利用第 5 章所介紹的「新式字體選用法」，若要選用 14.4pt 英文字體並把行距拉大成 20pt，指令為：

```
\fontsize{14.4}{20pt}\selectfont
```

如果字體再變回 12pt，行距縮小為 18pt，則必須再下指令

```
\fontsize{12}{18pt}\selectfont
```

以上之指令稍嫌複雜，為了簡化輸入動作，我們可以將指令定義於一巨集指令內，請見圖 14.1 的例子。

此一巨集指令是以 \TeX 之 `\def` 定義。巨集指令名字不能使用數字，因此我們將字級以變數方式填入。常用之字級包括 10pt, 10.95pt, 12pt 等，因此，巨集指令即以此等字級為主。以 12pt 字體為例，我們希望輸入指令簡化為 `\sz12`。若定義巨集時只設一項變數，輸入指令時數字 12 須置於括號內，變成 `\sz{12}`。為了進一步簡化，定義巨集時設定兩項變數，字級之十位數為第一項變數，個位數為第二項變數。如此，巨集指令可以簡化為 `\sz12`；其中，數字 12 之 1 為第 1 項變數，2 為第 2 項變數。

```

\newcount\fs
\def\sz#1#2{\fs=#1#2
\ifnum\fs=10\fontsize{10}{12.5pt plus.2pt minus .1pt}\selectfont
\else\ifnum\fs=11\fontsize{10.95}{17pt plus.3pt minus.2pt}\selectfont
\else\ifnum\fs=12\fontsize{12}{18.5pt plus.3pt minus .2pt}\selectfont
\else\ifnum\fs=14\fontsize{14.4}{20pt plus.4pt minus .3pt}\selectfont
\else\ifnum\fs=17\fontsize{17.28}{22pt plus.4pt minus .3pt}\selectfont
\else\ifnum\fs=20\fontsize{20.73}{28pt plus.5pt minus .3pt}\selectfont
\else\ifnum\fs=25\fontsize{24.88}{33pt plus.5pt minus .3pt}\selectfont
\else\ifnum\fs=09\fontsize{9}{11pt plus.2pt minus .1pt}\selectfont
\else\ifnum\fs=08\fontsize{8}{9.5pt plus.2pt minus .1pt}\selectfont
\else\ifnum\fs=07\fontsize{7}{9pt plus.2pt minus .1pt}\selectfont
\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi}

```

圖 14.1: 設定字級與行距之巨集指令

定義巨集時，我們同時也改變行距。例如，10pt 字體之行距為 12.5pt；伸縮彈性為正 0.2pt 與負 0.1pt。11pt 字體之行距為 17pt 加減伸縮彈性；12pt 字體之行距為 18.5pt，等等。若覺得以上的行距不妥，可以代入自己喜歡的數字。圖 14.1 亦定義 `\sz09`，`\sz08` 巨集指令。定義巨集指令之後，選用字體級數就變得很簡單。譬如，選用 14.4pt 之英文字只須鍵入 `\sz14` 指令即可。選用 17.28pt 字體，則鍵入 `\sz17`；12pt 字體，則鍵入 `\sz12`；餘此類推。如果是選用 8pt 之字體，則須鍵入 `\sz08`。請注意，輸入時不要忘了十位數之 0，否則將出現錯誤。

若每一篇文稿都要輸入圖 14.1 之巨集指令定義，顯然不方便。我們可以使用更簡單的辦法：將巨集指令儲存成一檔案。事實上，圖 14.1 之巨集指令已儲存為 `mymacro.tex`，安裝於 \TeX 系統內。因此，我們只須在全文設定區輸入：

```
\input{mymacro}
```

文稿內即可使用 `\sz12` 等指令改變字級。

排版時， \TeX 若遇到 `\input{...}` 指令，它會先讀取並處理該檔案之內容。因此，`\input` 指令之效果等於是將 `mymacro.tex` 之內容全部輸入於指令所在的位置。換句話說，`\input` 指令的目的也是在簡化輸入：把原本要放在文稿中內一大堆指令另置一處，以供隨時取用。安裝 $\text{cw}\TeX$ 時，巨集指令 `mymacro.tex` 置於 `c:\texmf\cwtex\examples` 檔案夾內，其內容僅含定義字級/行距之指令；但我們也可以將其他常用之巨集指令置於其中。

14.1.2 巨集指令與中文

上一小節所定義之巨集內不含中文字。事實上，一般的巨集指令內也可輸入中文字，但使用上須注意兩點：

- 巨集檔案須以 `.ctx` 為延伸檔名；
- 以 `\input` 或 `\include` 指令引入巨集檔案時，請勿加入延伸檔名。
- 巨集檔案須置於 `cwtext` 程式可找到之檔案夾內。

舉例言之，若巨集檔案之主檔名為 `mymacro`，`cwtext` 轉換中文時，若發現文稿內有 `\include{mymacro}` 或 `\input{mymacro}` 指令，它會在目前檔案夾 (`current directory`) 內尋找 `mymacro.ctx` 檔案。若檔案存在，即將之轉換為 `mymacro.tex`。接下來 `latex` 進行排版工作時，讀取的檔案是並非原始的 `mymacro.ctx`，而是中文碼已經轉換過的 `mymacro.tex`。因此，排版工作不會有問題。

請注意，輸入中文檔案時請勿加入延伸檔名。換言之，若文稿內所鍵入的指令為：`\include{mymacro.ctx}` 或 `\input{mymacro.ctx}`，排版時，`cwtext` 程式會試著尋找 `mymacro.ctx.ctx` 檔案；但因為檔案不存在，故不會作中文轉換動作。而 `latex` 如果直接讀取 `mymacro.ctx` 檔案，因為無法辨認其內之中文字，排版時即出現錯誤。另外要留意的是，`cwtext` 程式僅在目前檔案夾內尋找以 `.ctx` 為延伸檔名之檔案。巨集檔案若取名為 `mymacro.tex`，`cwtext` 找不到檔案，也不作中文轉換動作。

綜合以上所述，自行創造之巨集檔案若僅含英文，延伸檔名可取為 `.tex`，並且可置放於 `ETEX` 可以搜尋到的任何檔案夾內。反之，若巨集檔案內含有中文，請記得一定要以 `.ctx` 為延伸檔名，並且要與文稿檔案置於同一檔案夾內。

14.1.3 依條件處理之巨集指令

排版書籍時，單雙頁面可能須作不同的處理。舉例來說，某些圖表的寬度超過正文一行的長度，一個解決的辦法如下：若圖表是在雙數頁 (左頁)，可將圖表稍左移；若圖表是位於單數頁 (右頁)，則左邊切齊文字版面左沿，右邊則稍凸出一些。欲以巨集指令處理此一問題，我們需有一巨集指令能判斷本頁是單數頁還是雙數頁，並能依此作進一步的處理。David Carlisle 所寫的 `ifthen` 巨集套件可用來處理此一問題。

巨集套件 `ifthen` 提供指令 `\ifthenelse`，底下是一個簡單例子：

```
\usepackage{ifthen}
\newcommand{\chk}[2]{\label{#1}
\ifthenelse{\isodd{\pageref{#1}}}{\noindent\ignorespaces}%
{\noindent\hspace*{-#2\textwidth}\ignorespaces}}
```

在全文設定區引入巨集套件之後，接下來定義一巨集指令 `\chk`。此巨集指令使用了 `\ifthenelse` 指令，並且用了兩個參數。第一個參數是使用者填入之標誌 (label)，第 2 個參數設定左移之距離。因此，文稿內若輸入下列一行指令：

```
\chk{example}{0.1}{This is a test}
```

排版時， \TeX 首先在下指令處加入標誌 `example`，接下來利用 `\ifthenelse` 指令判斷此標誌是位於單數頁或雙數頁。若在單數頁，則取消行首內縮 (indent) 之動作；反之，若位於雙數頁，則以 `\hspace*{-0.1\textwidth}` 指令左移一點距離。不管是單數頁或雙數頁，最後都會排版 “This is a test”。

14.2 定義指令環境

除了巨集指令之外，我們也可以定義或修改指令環境。底下僅舉兩個例子簡單說明。第一個例子定義指令環境 `nm`，其功能與 \TeX 之 `enumerate` 指令環境類似，唯一不同的地方是加入 `\itemsep=-2pt` 設定，讓條列項之間距縮小 2pt。

```
\newenvironment{nm}{\begin{enumerate}\itemsep=-2pt}%
{\end{enumerate}}
```

由此例子可知，以 `\newenvironment` 定義新指令環境時，第一對大括號內置放指令環境名稱；接下來的兩對大括號，前一對大括號內含指令環境之定義指令；後一對則含結束之指令。

指令環境可包含參數，底下是一個簡單的例子。

```
\newenvironment{mymyp}[2][10mm]{\par\noindent\hspace*{#1}%
\begin{minipage}{#2\textwidth}}{\end{minipage}}
```

首先，指令環境名稱為 `mymyp`，其後之 `[2]` 表示此指令環境帶有第二項參數。但是，第一項參數是選擇性輸入 (optional)。如果使用此指令環境時僅輸入一項參

數值，則第一項參數將自動代入定義中之 10mm。舉例言之，經過以上之定義，若文稿中輸入下列指令：

```
\begin{mymargin}{0.8} ... \end{mymargin}
```

則鍵入文字內容將排版於迷你版面內，其寬度為正文行寬的 80%。排版時， \TeX 先結束上一段落，從版面左沿右移 10mm，再開始排版迷你版面。反之，若鍵入之指令為：

```
\begin{mymargin}[4mm]{0.8} ... \end{mymargin}
```

則迷你版面離正文版面左沿僅 4mm，而非內定值之 10mm。

14.3 更改內定之標題為中文

\TeX 原本是以排版英文稿件為主，但預留許多空間可以排版其他語文。 \TeX 的設計也留有許多彈性，以方便排版其他語文。譬如，以 `\caption` 指令排版表格標題時， \TeX 會自動加上 “Table” 一字並編上號碼。事實上，“Table” 是以 `\tablename` 定義。若以下列指令重新定義，

```
\renewcommand{\tablename}{ 表}
```

標題字即可改為中文。

為方便參考，表 14.1 列出所有之特定標題，及對應之英文標題。本表之指令中，`\enclname`，`\ccname`，`\headtoname`，`\headpagename` 等是用於 letter 文件類別；`\seename` 與 `\alsoname` 則用於索引。

14.4 計數器

\TeX 定義許多的計數器 (counter)，用以記錄章節、註解、方程式之編號。一般的情況下，計數器內的數字會自動加減；但必要時也可以重新設定。譬如，排版書籍時，若本頁為本章結束，下一頁要留為空白；則排版再一頁之前，頁碼計數器須先加 1。此外，我們也可以利用指令變更排版計數器數字之字體。

\TeX 所定義的計數器計有下列：

part	paragraph	figure	enumi
chapter	subparagraph	table	enumii
section	page	footnote	enumiii
subsection	equation	mpfootnote	enumvi
subsubsection			

以上各計數器之指令大多一看就了解其意義，唯一須解釋的是最右欄的四項計數器。其中，`enumi` 計數器記錄條列指令環境第一層項目數；`enumii` 記錄第二層，餘此類推。另外，`mpfootnote` 是用以記錄迷你版面指令環境內之註解數目。

欲重新設定某計數器之值，須使用 `\setcounter` 指令，譬如：

```
\setcounter{footnote}{20}
```

可將註解計數器定為 20。當 `TEX` 碰到下一個 `\footnote` 指令時，計數器將增加

表 14.1: 特定標題之指令與內定值

指令	英文標題
<code>\abstractname</code>	Abstract
<code>\appendixname</code>	Appendix
<code>\bibname</code>	Bibliography
<code>\ccname</code>	cc
<code>\chaptername</code>	Chapter
<code>\contentsname</code>	Contents
<code>\enclname</code>	encl
<code>\figurename</code>	Figure
<code>\headpagename</code>	Page
<code>\headtoname</code>	To (letter)
<code>\indexname</code>	Index
<code>\listfigurename</code>	List of Figure
<code>\listtablename</code>	List of Table
<code>\partname</code>	Part
<code>\prefacename</code>	Preface
<code>\refname</code>	References
<code>\seename</code>	see
<code>\alsiname</code>	see also
<code>\tablename</code>	Table

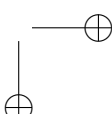
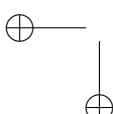
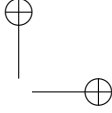
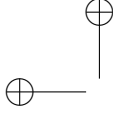
1, 因此註解編號變成 21。如果書籍是分章編排, 則利用同樣方法可以透過 `chapter` 計數器設定每章編號。例如, 某章為書本的第 8 章, 故文稿前端可加上:

```
\setcounter{chapter}{8}
```

不過, 實際排版時應儘可能讓 \LaTeX 自行設定章節編號。

除了直接設定計數器之數值之外, 我們也可以使用 `\addtocounter` 指令將現有的計數器之值加減某數。譬如, 若目前 `footnote` 計數器之值為 18, 則以下指令會將它變成 16:

```
\addtocounter{footnote}{-2}
```



15 有用的工具套件

TeX 主要是用來排版書籍與論文，但是它也可以用來幫助處理日常文書事務。譬如，它可以用來排版信函與大宗郵件；也可以用來排版投影片、習題解答等等。本章介紹一些常用的排版工具，除了信函、投影片之外，我們也將說明製作書籍索引的方法。

15.1 信函

TeX 有一套現成的巨集套件可排版信函。排版短文時，我們引用 `article` 文件類別；若要排版信函，我們可以直接引用信函文件類別 `letter`；也可以使用自行設計具個人風格之巨集指令。若是利用 `letter` 文件類別，排版信函之指令如下：

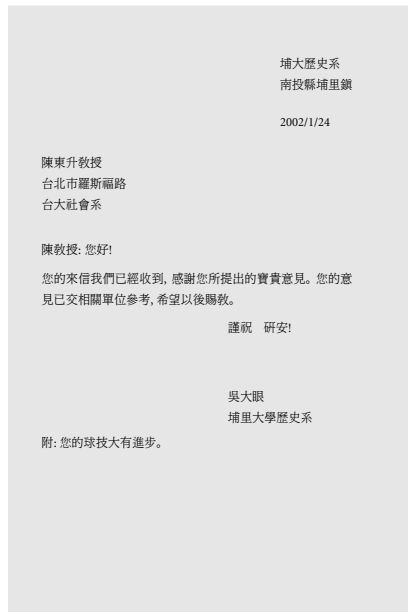
```
\documentclass[12pt]{letter}
\begin{document}
\begin{letter}{...}
...
\end{letter}
\end{document}
```

上例中，`[12pt]` 選項是指定使用 12pt 之英文字體。若不加此選項，則以內定之 10pt 字體編排。

TeX 的信函有特定的日期、地址、信頭、與信尾結語格式，但這些都可以調整。例如，信函內自動加上當天的日期，其形式如“March 19, 2002”。因為月份是以英文表示，用於排版中文信函並不適合。底下將說明修改的方法。

15.1.1 排版信函指令

為了說明方便起見，我們將信函分成前端、正文、結尾三部分。前端包括發信人地址、受信人姓名、地址與發信日期，結尾包括發信人姓名、職稱、附言等等。圖



```

\documentclass[12pt]{letter}
\address{ 埔大歷史系\\
        南投縣埔里鎮}
\signature{ 吳大眼\\
        埔里大學歷史系}
\renewcommand{\today}{\number\year/\%
        \number\month/\number\day}
\begin{document}
\fontsize{14.4}{22pt}\selectfont
\begin{letter}{ 陳東升教授\\
台北市羅斯福路\\
台大社會系}

\opening{ 陳教授: 您好!}
您的來信我們已經收到,
感謝您所提出的寶貴意見。
您的意見已交相關單位參考,
希望以後賜教。

\closing{ 謹祝 ~~ 研安!}
\ps 附: 您的球技大有進步。
\end{letter}
\end{document}

```

圖 15.1: 排版信函

15.1 之例子說明排版指令與結果, 檔名為 `letter1.ctx` , 置於 `cwTeX` 例子檔案夾 `c:\texmf\cwtex\examples` , 請試自行排版。

信函正文是以 `\begin{letter}{...}` 起頭, 大括號內輸入受信人姓名、地址等。如果受信人姓名及地址要留為空白, 請鍵入 `\begin{letter}{\ }` , 否則排版時會出現錯誤。接著, 以 `\opening{...}` 指令排版受信人姓名及致敬詞句; 其下即輸入信函正文。最後以 `\closing{...}` 指令加入結尾祝福言辭。

一個檔案中可以同時排版數封信函, 每封信函都須置於 `letter` 指令環境內。若在全文設定區以 `\address` 指令宣告發信人之地址, 此地址會出現在每一封信函前端。本例中, 全文設定區內有三道指令, `\address` 指令是用以排版發信人地址。地址若有兩行以上, 則以 `\\` 換行指令隔開。其次, `\signature` 指令用以排版信函結尾處之發信人姓名及職稱。若有兩行以上的內容, 也須以換行指令 `\\`

隔開。排版之後，發信人地址之下會自動排出當天日期。但日期是以英文格式出現，並不適用於中文信函。要改變日期格式，最簡單的方法是在全文設定區重新定義。本例重新定義 `\today` 指令，排版結果日期格式將變成“2002/1/24”。若把指令中的“/”右斜線符號改成“.”，排版結果將變成“2002.1.24”。

信函正文可以使用一般 \LaTeX 指令編排。信函結尾處除了發信人姓名、職稱之外，可以用 `\cc{...}` 指令列出其他受信人的姓名。另外，`\ps` 指令則用以排版附言。如果隨函附有其他文件或物品，我們可以用 `\encl{...}` 加以說明。若使用 `\encl{...}` 指令，排版時 \LaTeX 自動加上“encl:”符號；使用 `\cc{...}` 指令，則自動加入“cc:”符號。但是，`\ps` 指令並不加上任何符號，因此用來排版中文附言很方便。以上之英文標題都可以重新定義，請見 14.3 節之說明。

15.1.2 設計個人信頭標誌

如果常有信件來往，我們可以設計一信頭標誌置於信函第一頁上端。排版信函時，我們使用的是 `letter` 文件類別。此一巨集套件之設計原就允許加入個人信頭標誌。一旦設計好信頭標誌，要引入信函內並不困難。

若 \LaTeX 系統是安裝於硬碟 `c:`，則在 `c:\texmf\tex\latex\base` 檔案夾下可找到 `letter.cls` 文件類別檔案。欲在信函上加上個人信頭標誌，我們應另寫一個文件類別檔案，假設取名為 `sletter.cls`。此檔案前端應該先引用 `letter` 文件類別，其下再以指令設計個人信頭標誌。如此，排版時仍可使用原 `letter` 文件類別之指令。圖 15.2 上端為信頭標誌，其下為指令內容。前 5 行指令定義文件類別檔案名為 `sletter`，並引用 `letter` 文件類別。為了易於區分起見，第 6 行留為空行。第 7–13 行為個人信頭標誌之定義。為了方便設計及修改，第 7 行先以 `\newsavebox` 設定使用文字方格，取名為 `\ltrhead`；第 8–13 行以 `\sbox` 指令將信頭標誌指令置於文字方格中，

若只是要設計簡單的英文信頭標誌，指令很簡單。本例中，信頭左方是粗體字 **ABC Company**，右方是地址與電話，底下再畫出一條直線。排版時，我們使用兩個 `\parbox` 指令，第一個容納整個信頭標誌，第二個用於排版地址與電話。第一個迷你版面之寬度為 15cm，這是信頭標誌之寬度。公司名稱選用較大之粗黑字體排版；地址與電話則占用版面寬度的 30%。因為 `\parbox` 指令加入 `[b]` 選項，因此地址與電話之迷你版面下沿將對準公司名稱之基線。`\hfill` 指令將公司名稱與地址迷你版面推向版面的兩邊。

ABC Company

123 xyz Street, Taipei
(02)2987-6543

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{sletter}
\DeclareOption*{\passOptionsToClass{\CurrentOption}{letter}}
\ProcessOptions
\LoadClass[12pt]{letter}

\newsavebox{\ltrhead}
\abox{\ltrhead}{\parbox{15cm}
{{\Large\bf ~~ABC Company}\hfill
\parbox[b]{.3\textwidth}{\small
123 xyz Street, Taipei\[-2pt]
(02)2987-6543}\}}
\rule[4pt]{15cm}{.6pt}}
\renewcommand{\ps@firstpage}
{\setlength{\headheight}{3cm}\setlength{\headsep}{.4cm}%
\renewcommand{@oddhead}{\hspace*{-.6cm}%
\raisebox{1cm}{\usebox{\ltrhead}}}}
\renewcommand{\ps@headings}
{\renewcommand{@oddhead}{}}
\renewcommand{@oddfoot}{}}
\renewcommand{@evenhead}{@oddhead}
\renewcommand{@evenfoot}{@oddfoot}
\providecommand{@evenhead}{}\providecommand{@oddhead}{}}
\providecommand{@evenfoot}{}\providecommand{@oddfoot}{}}
\pagestyle{headings}

```

圖 15.2: 個人信函標誌

信頭標誌通常是出現在信函的第一頁，我們須在指令中宣告標誌本身的高度。第 14 行指令開始是重新定義信函首頁之信頭標誌，我們利用 `\headheight` 指令定義信頭標誌之高度為 3cm，以 `\headsep` 指令定義標誌與信函內文之間距的 0.4cm。信函第 2 頁（偶數頁）開始不再列印信頭標誌，故將 `@evenhead` 定為空白。若信頭之外，信函下端也有特別之標誌，我們也可以依類似方法設計。詳細說明，請參見 [Kopka and Daly \(1995\)](#)，附錄 A。

檔案 `sletter.cls` 製作完成之後，請移入 `c:\localtexmf\tex\latex` 下之檔

案夾內, 或者, `c:\tex` 檔案夾下子目錄內。之後, 請更新檔案資料庫, 讓排版程式可以找到此檔案。安裝 MiKTeX 者, 請執行下列指令:

```
c:\>initexmf -u
```

安裝 fpTeX 者, 請執行指令:

```
c:\>mktexlsr
```

排版信函時, 第一行指令應為:

```
\documentclass{sletter}
```

信函內容之輸入方法則仿照圖 15.1 之例子。在製作文件類別檔案 `sletter.cls` 時, 第 5 行以 `\LoadClass` 指令引入 `letter.cls`, 其中已加入 `[12pt]` 選項, 故以上使用文件類別指令時不須再加入字體大小之選項。請特別注意, 個人信頭標誌內通常已含有發信人地址, 因此信函內容不能再使用 `\address` 指令。若加入此指令, 排版後信函首頁將不會出現個人信頭標誌。

信函排版之後, 首頁上方即出現自行設計之信頭標誌。若信函長度超過一頁, 第 2 頁開始, 個人標誌不再出現, 每頁上方僅列出頁碼、收信人姓名、日期等。
`c:\texmf\cwtex\examples` 檔案夾內存有 `sletter.cls` 文件類別及 `letter2.ctx` 信函例子檔案, 有意自行設計信函標誌者, 請試排版。

15.1.3 中文信頭標誌

如果我們的目的只是製作一個簡單的英文信頭標誌, 則仿照圖 15.2 之指令應已足夠。若信頭含有中文字, 設計方法相同, 但處理上較為複雜。假設要我們設計一個如圖 15.3 所示之信頭標誌。因為版面較複雜, 我們先以一 \TeX 檔案編排此標誌。圖 15.3 下方為指令內容。此檔案取名 `ntueltr.ctx` 置於 `cwTeX` 例子檔案夾內, 請自行取出排版。

我們將信頭標誌設計為一文字方格。實際上, 此例使用兩個文字方格指令, 第一個方格為 `\headlogo` 用以存放標誌左上方的三道斜線, 第二個文字方格為 `\ltrhead`, 存放整個信頭標誌。整個信頭標誌首先引入斜線標誌 `\headlogo`, 其右為「台大經濟系」, 底下為地址。右方之姓名、職稱、電話號碼等則利用迷你版面指令編排。標誌底下加上一條橫線, 長度略大一些。

/// 臺大經濟系

台北市徐州路21號

吳聰敏

台大經濟系教授

email: ntut019@ccms.ntu.edu.tw

Tel: (02)2351-9641

```

\documentclass{article}
\begin{document}
\newsavebox{\headlogo}
\newsavebox{\ltrhead}
\sbbox{\headlogo}{\parbox[b]{8mm}{
\fontfamily{cmss}\fontshape{sl}\fontsize{17.28}{20}\selectfont III}}
\sbbox{\ltrhead}{
\begin{minipage}{16cm}
\parbox[b]{.54\textwidth}{
\usebox{\headlogo}
{\LARGE\ctxfbb 臺大經濟系}\[4pt]
{台北市徐州路21號}}
\parbox[b]{.3\textwidth}{
吳聰敏\ 台大經濟系教授}
\hspace*{-.03\textwidth}\rule[4pt]{.95\textwidth}{.6pt}
\hspace*{.46\textwidth}\parbox{.4\textwidth}{
\raggedleft
email: ntut019@ccms.ntu.edu.tw\
Tel: (02)2351-9641}
\end{minipage}}
\usebox{\ltrhead}
\end{document}

```

圖 15.3: 中文個人信頭標誌

信頭標誌設計完成之後，須經過處理，才可引入 `ntueltr.cls` 檔案內。圖 15.3 所示之檔案名為 `ntueltr.ctx`，請將之複製於 `c:\xtemp` 檔案夾內，執行 `cwtex` 將中文字轉換成 TeX 格式：

```
c:\xtemp>cwtex -i ntueltr
```

以上指令中加入 `-i` 選項的目的，是要使中文字之定義指令（亦即 `cinput.tex` 之內容）直接置於 `ntueltr.tex` 檔案前端。

以文字編輯軟體讀取 `ntueltr.tex`，檔案前半部分為中文字體定義指令，後半部分為中文信頭設計指令。檔案前半部分之指令又分為兩部分，一為中文字距指令：`\z`，`\Z`，`\zZ`；其次是中文字體指令。以明體字為例，中文字型檔案之檔名為 `\MaQ`，`\MbQ`，... 等等。其中，`MaQ` 代表 10 點明體字之第 0 個字型檔。若為粗黑體字，檔名將為 `\cBaQ`。每一個中文字在其所屬字型檔案中所占位置，其順序是以 `\cH` 指令指定。譬如，若某中文字位於第 171 個位置，指令為 `\cH171`。

將 `\documentclass` 與 `\begin{document}` 兩行指令，以及最後兩行指令前端加上註銷指令或者直接刪除。經過以上之處理，其餘指令即可引入 `ntueltr.cls` 檔案內。此檔案其他部分之定義與圖 15.2 所示之指令類似。為方便參考，我們將 `ntueltr.cls` 檔案內容陳列於下。

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{ntueltr}
\RequirePackage{ifthen}

\DeclareOption*{\passOptionsToClass{\CurrentOption}{letter}}
\ProcessOptions
\LoadClass[12pt]{letter}
\voffset=-2.5cm
\textheight=8.6in

\renewcommand{\today}{\number\month-\number\day-\number\year}

%% input from ntueltr.tex: start
\providecommand{\z}{\hspace 0.0pt plus0.2pt minus0.1pt}
\providecommand{\Z}{\hspace 1.2pt plus0.4pt minus0.2pt}
\providecommand{\zZ}{\hspace 3.6pt plus1.2pt minus0.8pt}
\providecommand{\cH}{\char}
\providecommand{\MaQ}{\fontfamily{cwM0}\fontseries{m}\selectfont}
\providecommand{\cBaQ}{\fontfamily{cwBB2}\fontseries{m}\selectfont}
\providecommand{\cBaQ}{\fontfamily{cwBB0}\fontseries{m}\selectfont}
\providecommand{\cBbQ}{\fontfamily{cwBB1}\fontseries{m}\selectfont}
\providecommand{\cBhQ}{\fontfamily{cwBB7}\fontseries{m}\selectfont}
\providecommand{\MbQ}{\fontfamily{cwM1}\fontseries{m}\selectfont}
\providecommand{\MfQ}{\fontfamily{cwM5}\fontseries{m}\selectfont}
\providecommand{\McQ}{\fontfamily{cwM2}\fontseries{m}\selectfont}
\providecommand{\MdQ}{\fontfamily{cwM3}\fontseries{m}\selectfont}
\providecommand{\MiQ}{\fontfamily{cwM8}\fontseries{m}\selectfont}
\providecommand{\MhQ}{\fontfamily{cwM7}\fontseries{m}\selectfont}
% \documentclass{article}
% \begin{document}
\newsavebox{\headlogo}
\newsavebox{\ltrhead}
\sbox{\headlogo}{\parbox[b]{8mm}{

```

```

\fontfamily{cms}fontshape{sl}%
\fontsize{17.28}{20}\selectfont III}
\sbx{\ltrhead}{
\begin{minipage}{16cm}
\parbox[b]{.54\textwidth}{
\usebox{\headlogo}
{\LARGE{\cBcQ\cH67}\z{\cBaQ\cH215}\z{\cBcQ\cH37}\z%
{\cBbQ\cH200}\z{\cBhQ\cH205}}\l[4pt]
{\MaQ\cH171}\z{\MaQ\cH148}\z{\MbQ\cH13}\z{\MfQ\cH15}\z{\MbQ\cH8}\z%
{\McQ\cH152}21\Z{\McQ\cH85}}}
\parbox[b]{.3\textwidth}{
{\MdQ\cH216}\z{\MiQ\cH28}\z{\MfQ\cH220}\l\ {\MaQ\cH171}\z{\MaQ\cH215}%
\z{\McQ\cH37}\z{\MbQ\cH200}\z{\MhQ\cH205}\z{\MbQ\cH96}\z{\MfQ\cH164}}
\hspace*{-.03\textwidth}\rule[4pt]{.95\textwidth}{.6pt}
\hspace*{.46\textwidth}\parbox{.4\textwidth}{
\raggedleft
email: ntut019@ccms.ntu.edu.tw\
Tel: (02)2351--9641}
\end{minipage}
% \usebox{\ltrhead}
% \end{document}
%% input from ntueltr.tex: end

\renewcommand{\ps@firstpage}
{\setlength{\headheight}{3cm}\setlength{\headsep}{.4cm}%
\renewcommand{\oddhead}{\hspace*{-.6cm}%
\raisebox{1cm}{\usebox{\ltrhead}}}}
\renewcommand{\ps@headings}
{\renewcommand{\oddhead}{}}
\renewcommand{\@oddfoot}{}
\renewcommand{\@evenhead}{\@oddhead}
\renewcommand{\@evenfoot}{\@oddfoot}
\providecommand{\@evenhead}{} \providecommand{\@oddhead}{}
\providecommand{\@evenfoot}{} \providecommand{\@oddfoot}{}
\pagestyle{headings}

```

以上過程中三個檔案: ntueltr.ctx , ntueltr.tex 與 ntueltr.cls 都可以在 cwTeX 例子檔案夾內找到, 請自行參考。檔案夾內另有 letter3.ctx 測試檔, 可排出 ntueltr.cls 之標誌。如果你自行設計信頭標誌, 檔案完成之後, 建議移入

```
c:\localtexmf\tex\latex\mytex
```

檔案夾內, 若者 c:\tex 下自選之檔案夾內。之後, 請更新檔案資料系統。MiKTeX 使用者, 請執行:

```
c:\xtemp>initexmf -u
```

fpTeX 使用者, 請執行:

```
c:\xtemp>mktextlsr
```

15.2 大宗信函

欲編排較複雜的大宗信函, 可使用 Mike Piff 所寫的 `textmerg`。排版大宗信函須準備兩份檔案, 一為信函內容, 一為地址檔案。信函正文之格式類似一般的 `letter` 文件類別, 但其中不輸入收信人地址與收信人尊稱。收信人地址須另輸入成一單獨檔案, 其中存放所有受信人之姓名、地址、稱呼或其他相關資訊。

圖 15.4 中, 左邊為信函內容, 右邊為地址檔內容。本例中, 每一封信將自地址檔案中取用 4 項資訊: 收信人頭銜、地址、姓氏; 最後一項資訊內容為空白。我們特別定義一空白資訊, 目的是讓地址檔案內所排列之資訊易於區分。此例中, 地址檔取名為 `address.ctx`, 其中有三位收信人。假設信函檔名為 `bulkmail.ctx`, 我們在其內以 `\Fields{...}` 指令定義地址檔案內 4 項資訊之指令名稱, 分別為: `\Title` (收信人頭銜), `\Add` (地址), `\Surname` (姓氏), 與 `\en` (空白)。最後的 `\en` 代表空白資訊。以上之指令名稱可任取。本例中, 地址名稱並不長, 因此全部輸入為一行。若地址較長, 我們可以將之拆為兩部分, 譬如, 門牌與街名以 `\street` 表示, 城市以 `\city` 表示。

信函檔案內首先引入 `textmerg` 巨集套件。其次, 為方便處理地址檔案, 我們加入 `\include{address}` 與 `\includeonly{}` 兩道指令。本例中, 信函與地址皆為中文。TeX 編排信函時, 須自地址檔中讀取收信人相關資訊。因此, 開始編排之前須先使用 `cwtex` 轉換信函正文與地址檔案內之中文字。若 `bulkmail.ctx` 與 `address.ctx` 皆置於同一檔案夾內, 而且前一檔案內有 `\include{address}` 指令, 則 `cwtex` 在轉換 `bulkmail` 檔案時, 會自動找出 `address.ctx`, 同時轉換檔案內之中文字。請注意, 地址檔案之延伸檔名必須是 `.ctx`。

以 `\include` 指令轉換地址檔時, 請在全文設定區加入 `\includeonly{}` 指令, 此可避免 TeX 把地址檔內容也編排於信函內。本例中, `address.ctx` 地址檔最前面有 3 行說明文字。地址檔案經轉換中文之後, 排版 `bulkmail` 之前, 請將這行刪除, 因為 `textmerg` 巨集套件無法處理這幾行。

大宗信函與地址檔案之編排控制是以 `\Merge{address.tex}{...}` 指令為之。其中, 第一圈大括號內之 `address.tex` 是經過轉換、而且已刪除前端說明

```

\documentclass{ntueltr}                % example of add file
\usepackage{textmerg}                 % file: address.ctx
\signature{ 吳聰敏\                  %
  台大經濟系}                         主任
\includeonly{}                        台中商專\ 台中市三民路
\begin{document}                       簡
\include{address}                      主任
\Fields{\Title\Add\Surname\en}        糖業研究所\ 台南市生產路
\Merge{address.tex}{%                 蕭
\begin{letter}{\Surname\Title\  \Add}  主任
\opening{\Surname 主任: ~ 您好!}     高雄工專\ 高雄市建工路
\fontsize{12}{22pt}\selectfont        趙
陳中文先生擬申請貴校教職,
本人很高興推薦之...
敬請\Surname 主任惠予考慮。
崙此, ~ 順祝
\closing{ 研安!}
\end{letter}}
\end{document}

```

圖 15.4: 大宗郵件 — textmerg 巨集套件

文字之地址檔案。第二圈大括號內則為 `letter` 指令環境之內容。`textmerg` 巨集套件可以和上一小節所製作之個人信頭標誌一起使用。事實上, 圖 15.4 的例子即引用了自行設計的信頭。

以上說明以 `\include` 指令轉換地址檔案的方法。若不使用此方法, 也可以自行轉換, 方法如下。地址檔案輸入完成之後, 若檔名為 `address.ctx`, 且置於 `c:\xtemp` 檔案夾內, 首先執行以下指令:

```
c:\xtemp>cwtext -- address
```

請注意要加上 `--` 選項, 其目的是避免將定義中文字體指令之 `cinput.tex` 檔案引入 `address.tex` 內。轉換地址檔案之後, 須緊接著轉換信函檔案 `bulkmail.ctx`, 中間不可處理其他文稿:

```
c:\xtemp>cwtext + bulkmail
```

請特別注意加入選項 `+`, 其目的是將轉換 `address.ctx` 與 `bulkmail.ctx` 兩項檔

案後所產生之 `cinput.tex` 合併。轉換完畢，若 `address.tex` 檔案前端仍有以 % 開頭之說明指令，請將之刪除。

15.3 固定格式標籤

大宗郵件的地址通常先列印於自粘式標籤上，再貼上郵件。欲排版固定格式之標籤，可使用 `labels` 巨集套件，這是由 Sebastian Rahtz, Leonor Barroca 與 Grant Gustafson 所合力創作。不管是大宗郵件之地址、唱片或錄音帶標籤之標示等，都可以用此巨集套件編排；標籤格式很容易自行設定。

市面上可以買到各種品牌之自粘標籤，格式形形色色。以美國 Avery 牌子為例，產品編號 5260 之標籤在一張 letter size 紙張上印有 3 欄 10 行共計 30 張之空白標籤。產品編號 5360 則是 3 欄 7 行，共計 21 張空白標籤。列印標籤時，紙張大小尺寸須控制精準，否則內容可能印到標籤之外。其他品牌之標籤可能會標示與某一種 Avery 格式是相容的。

台灣則同時通行 letter size 與 A4 兩種規格之紙張尺寸。如果你使用的是 letter size 紙張，請在文件類別指令中加上 `letterpaper` 選項，若使用的是 A4 紙張，請加入 `a4paper` 選項。選用 `a4paper` 時，`labels` 巨集套件自動設定 3 欄/7 行格式。若選用 `letterpaper`，內設值為 3 欄/8 行格式；但此內設值可自行更改。

如圖 15.5 所示，`labels` 巨集套件提供 `labels` 指令環境，紙張與標籤大小之控制指令須置於全文設定區。本例中，標籤為 3 欄 7 行，計 21 張。每一張標籤內文字排版位置可自行控制，譬如，`\LeftBorder` 指令用以調整標籤內左方之空白。標籤之內容輸入於 `labels` 指令環境內，格式很簡單。如圖 15.5 所示，若某一張標籤內容有三行文字，直接鍵入即可；標籤之間以空行分隔。本例為了簡化起見，僅輸入兩項。

標籤紙面上下方之空白可以用 `\TopBorder`，`\BottomBorder` 兩項指令調整。本例中，分別設定為 9mm 與 2mm。以 A4 紙張為例，`\paperheight` 等於 29.7 公分。由此高度減去 `\Topborder`，再減去 `\BottomBorder` 之後，除以 `\LabelRows` 即可算出每一小標籤之高度。實際排版時，須來回測試幾次，先試列印於普通紙上。得到正確結果之後，最後再列印於自粘標籤紙上。

以上例子中，標籤文字直接輸入 `TeX` 檔案內，但我們也可以將標籤內容全部輸入於單獨檔案中，再引入排版文稿。輸入標籤內容時，各單項之間以空白行分

```

\documentclass[12pt,a4paper]{article}
\usepackage{labels}
\LabelCols=3%      Number of columns of labels per page
\LabelRows=7%      Number of rows of labels per page
\LeftBorder=8mm%   Space added to left border of each label
\RightBorder=8mm%  Space added to right border of each label
\TopBorder=9mm%    Space to leave at top of sheet
\BottomBorder=2mm% Space to leave at bottom of sheet
\begin{document} % End of preamble
\fontsize{12}{13.5pt}\selectfont
\begin{labels}
台大經濟系
台北市徐州路 21 號
0)2351-5468

蕭耀基
糖業研究所
台南生產路 54 號
\end{labels}
\end{document}

```

圖 15.5: 固定格式標籤

隔。多行之空白視同一行空白處理。標籤內容中若還有註銷符號 `%`，該符號後面文字即不加處理。若標籤內容存放於 `names.dat` 檔案內，排版指令如下：

```

\usepackage{labels}
\includeonly{}
\begin{document}
\include{names}
\labelstyle{names.tex}
\end{document}

```

請注意，`\labelfile` 指令並不須置於 `labels` 指令環境內。

本例中，我們在第 2 行與第 4 行分別使用 `\includeonly` 與 `\include` 指令，其目的是將地址檔案內之中文字轉換為 \TeX 指令。其中，`\include` 指令僅能出現於 `document` 指令環境內，其功能視全文設定區 `\includeonly` 指令之內容而定。若後一指令之參數內容空白（如本例所示），則 `\include` 指令等於是 `\clearpage`，沒有其他作用。換言之， \TeX 將結束本頁之排版，將遺留之浮動版面圖表全部排出，準備排版下一頁。

```

\documentclass[11pt,letterpaper]{article}
\usepackage{labels}
\LabelRows=10
\numberOflabels=30
\woffset=1.5cm
\LeftBorder=1.2cm
\TopBorder=1.3cm
\BottomBorder=1.2cm
\begin{document}
\fontsize{11}{12pt plus.6pt minus.4pt}\selectfont
\addresslabel{ 台大經濟系\\
台北市徐州路21號\\
(O)02-2351-5468\\
(F)02-2351-1826}
\end{document}

```

圖 15.6: 重覆內容之標籤

第4行指令 `\include{names}` 指令指示 `cwtex` 程式搜尋硬碟內 `names.ctx` 檔案。若找到檔案，則將其內之中文字轉換為 `TEX` 指令，以讓 `LATEX` 可以排版出標籤。請特別注意，`\include` 指令僅能輸入主檔名，副檔名不可輸入，否則會發生錯誤。

除了排版收信人地址外，`labels` 巨集套件也可以用來排版發信人地址。在西式信封上，信函左上方填寫送信人地址。經常寫信的人可以用自粘標籤排出一整頁姓名/地址，寫完信後，撕下一標籤粘上信封即可寄出。圖 15.6 是排版送信人地址的例子，此例子適用於 letter size 紙張之 3 欄 10 行之標籤格式，也就是 Avery 品牌產品編號 5260 之格式。因為一頁紙面上將排出 30 張標籤，而且每一標籤之內容完全相同，故全文設定區以下列指令設定重複列印 30 份標籤：

```
\numberOflabels=30
```

標籤內容是以 `\addresslabel` 指令排版，每行末端須自行加上換行指令 `\\`。請注意，`\addresslabel` 僅須鍵入一次，排版時程式會自動複製於 30 份標籤之內。如果要列印兩整張的標籤，上述指令中之 30 須改為 60。

15.4 投影片

論文報告或成果展示經常使用投影片。`LATEX` 有幾套排版投影片的巨集套件，本

書第2版介紹其中的 seminar 文件類別, 作者是 Timothy van Zandt。本版則改介紹 Foiltext 巨集套件, 原因之一是 Foiltext 使用較方便, 功能也相當不錯。更重要的是, 目前單槍投影機日漸普及, 有逐漸取代傳統投影機的趨勢。利用 Foiltext 與 pdfscreen 巨集套件, 並以 pdf \LaTeX 編排, 可製作出類似 Power Point 效果之 PDF 檔案, 由單槍投影機放映。

15.4.1 FoilTeX 巨集套件

一般人自行以 \LaTeX 排版之投影片, 最大的問題是字體太小, 聽講者不易看清楚投影片上的文字。另外, 若投影片有橫排也有直排, 編排也不容易。使用 Foiltext 巨集套件, 這些問題都很容易解決。此一套件之功能甚佳, 而且使用方便, 作者是 Jim Hafner。

簡單來說, 投影片之內容可分標題頁及各張投影片兩部分。整份文稿之結構如圖 15.7 所示。Foiltext 提供的是 foils 文件類別, 故直接以 `\documentclass` 指令引用之。本例中, 我們加上 `dvips` 選項, 以便能處理直排與橫排之投影片。依內定值, 投影片將以 20pt 之 sans serif 字體排版, 故第 3 行指令加入 `\ctxfbb` 中文字體指令, 以匹配英文字體。

如果你覺得 20pt 字體大小不適當, 可在 `\documentclass` 指令中加入 17pt, 25pt, 或 30pt 選項, 以選定基本字體之尺寸。此外, 若加入 `headrule` 選項, 可在每張投影片上方畫出橫線, `footrule` 選項則在投影片下方畫直線。因為字體較大, 若強迫每行文字右沿對齊, 可能會出現字距太大的現象, 故我們在全文設定區加上 `\raggedright` 指令, 取消右沿對齊之設定。

因為是中文投影片, 故我們以 `\linespread` 指令加大行距為原來的 1.2 倍。Foiltext 文件類別提供 `\MyLogo` 指令, 以排版投影片之標誌。圖 15.7 例子中, 此指令定義「經濟成長」4 個字。這 4 個字會出現在每張投影片的左下角, 右下角則排出頁碼。我們也可以利用 `\includegraphics` 指令將圖形引入投影片標誌內。依內定值, 投影片採直排 (portrait) 格式。欲採橫排格式, 文件類別指令內請加入 `landscape` 選項。若整篇文章是採直排, 個別的投影片仍可橫排, 底下將會說明。

每一張投影片是以 `\foilhead{...}` 指令起頭, 其下空出 18pt 距離, 再開始排版投影片內容。若認為此一間距不適當, 可在 `\foilhead` 指令中加入設定指令。例如, 欲縮小間距 2 公分, 指令為:

```

\documentclass[dvips]{foils}
\usepackage{graphicx,color}
\ctxfbb
\raggedright
\MyLogo{ 經濟成長}
\linespread{1.2}

\title{ 日本殖民統治與台灣的經濟成長}
\author{...}
\date{...}
\begin{document}
\maketitle

\foilhead[-2cm]{\textcolor{blue}{ 目錄}}
...

\rotatefoilhead[-2cm]{\textcolor{blue}{ 台灣長期 GDP 統計}}
...

\end{document}

```

圖 15.7: 以 Foiltex 製作投影片

```
\foilhead[-2cm]{...}
```

若原設定是以直排方式排版, 現某張投影片要採橫排, 應使用之指令為:

```
\rotatefoilhead[-2cm]{...}
```

反之, 若原設定為橫排, 以上指令將使本張投影片改為直排。

15.4.2 數學式

為方便排版數學文稿之定義、定理、與證明, Foiltex 提供幾個特別設計之指令環境, 包括: `Definition`, `Theorem`, `Lemma`, `Corollary`, 與 `Proposition`。請注意, 第一個字母都是大寫, 以與 \LaTeX 原有之 `theorem` 巨集套件有所區別。以上之指令環境若加上 * 號, 如 `Theorem*`, 則定理不會自動編號。定理之證明可以用 `Proof` 指令環境編排。以 `Theorem` 指令為例, 排版後定理前端會自動加上 `Theorem` 英文字。同樣的, 以 `Proof` 指令環境排版時, 證明前端會自動加上 `Proof` 英文字。若希望定理前端出現中文「定理」兩字, 則應使用 `theorem` 巨集套件, 請參見 9.4 節 (頁 171) 之說明。

投影片之內容主要是以 `sans serif` 字體編排, 爲了方便數學式以粗體字編排, FoilTeX 特別提供 `\bm` (代表 `bold math`) 指令。譬如, 下列之指令

$$\mathbf{\$x^n+y^n=z^n\$}$$

可使數學式以粗體字排出。此外, 我們也可以使用 `boldequation` 指令環境使整行或數行之數學式以粗體字排版。若方程式不想自動編號, 則使用 `boldequation*` 指令環境。

15.4.3 圖形與彩色

在 \TeX 中, 我們可使用浮動圖表指令環境 `figure` 與 `table` 安置圖表。不過, 排版投影片時, 圖表應該是置於下指令之位置, 而不應移往版面上方或下方。故在 FoilTeX 中, 若使用 `figure` 與 `table` 指令環境, 圖表並不會移往他處, 而是直接排版於指令所在的位置。最後, 欲調整圖表本身與標題文字之間距, 我們可以使用 `\abovecaptionskip` 與 `\belowcaptionskip` 指令, 請參見第 10.4.2 節 (頁 205)。

除了引入彩色圖形之外, \TeX 的彩色指令可以直接用於投影片內。圖 15.7 例子中, 我們使用 `\textcolor` 指令將每張投影片之標題以藍色排版。讀者可試排版 `cw\TeX` 例子檔案夾內之 `foil-1.ctx`。

15.5 Power Point 樣式投影片

使用上一節所介紹之 FoilTeX 巨集套件, 排版結果印出之後, 以影印機複印於投影片上, 可經由傳統投影機放映出來。不過, 單槍液晶投影機發明之後, 目前已有逐漸取代傳統投影機之趨勢。單槍投影機的功能是將電腦顯示器上之畫面直接放映出來。在 Windows 作業系統上, Microsoft 的 Power Point 就是一套有名的簡報軟體。

在 \TeX 系統下, 我們也可以製作出類似 Power Point 功能之投影片。具體言之, 我們利用 `pdf\TeX` 將文稿排版成 PDF 格式, 經由 Acrobat Reader 軟體顯示, 再透過單槍投影機放映, 其效果類似 Power Point。如果簡報內容含複雜數學式或圖表, 則由此方法所製作之簡報, 其效果絕對不亞於 Power Point。

我們如何製作 PDF 投影片呢? 事實上, 利用上一節所介紹之 FoilTeX, 將投影片文稿直接以 `pdf\TeX` 排版, 即可產生 PDF 檔案。此檔案由 Acrobat Reader 顯

```

\documentclass[landscape]{foils}
\usepackage{graphicx,color}
\ctxfbb
\MyLogo{ 經濟成長}
\linespread{1.2}

\usepackage{pause}
\usepackage[screen,nopanel,gray]{pdfscreen}
\bottombuttons
\screensize{6.25in}{9in}
\margins{.4in}{.4in}{.4in}{0.8in}
\overlay{cwback.pdf}

\title{ 日本殖民統治與台灣的經濟成長}
\author{...}
\date{...}
\begin{document}
\maketitle

\foilhead[-2cm]{\textcolor{blue}{ 目錄}}
...

\end{document}

```

圖 15.8: 以 pdfscreen 製作 PDF 投影片

示,經由單槍投影機播出,即為一張一張效果不錯之投影片。不過,由以上方法所產生的投影片並無特別的視覺效果。舉例來說,如果在某張投影片上我們是利用 `itemize` 指令環境排版 4 點事項,投影片放映時無法將這 4 點分項依序列出,而是同時放映出來。但若利用巨集套件,即可作出以上的效果。此外,若希望投影片具有溶入淡出之效果,也可以透過巨集套件完成。

CTAN 上有幾套加強 PDF 檔案放映效果的巨集套件,本節僅介紹其中較簡單的一套: `pdfscreen`, 作者為 C.V. Radhakrishnan。

15.5.1 pdfscreen 巨集套件

`pdfscreen` 巨集套件若與上一節之 `Foiltex` 配合使用,可排版出效果絕佳之 PDF 投影片。此巨集套件是以 Sebastian Rahtz 所寫的 `hyperref` 巨集套件為基礎,主要的功能事實上都是由後者完成。使用方法很簡單,將上一節圖 15.7 之例子稍作修改,並引入相關的 `pdfscreen` 指令即告完成,圖 15.8 是一個例子。

首先，第 1 行指令不再加入 `dvips` 選項，原因是投影片必須以 `pdf \LaTeX` 編排，直接產生 PDF 檔案，不再使用 `DVIPS` 程式。事實上，引入 `pdfscreen` 巨集套件之後，若以 `latex` 排版，將會出現錯誤訊息。單槍投影機之投影片內容通常是採橫式編排，故我們加入 `landscape` 選項。本例中，7–12 行是新加入之指令。第 7 行指令引入 `pause` 巨集套件，這是 `PPower4` 巨集套件的一部分，底下將說明其功能。第 8 行引入 `pdfscreen` 巨集套件，第 1 個選項 `screen` 指示產生顯示器放映之格式。此選項若改用 `print`，則產生可列印之 PDF 檔案。另外，`gray` 選項用以設定畫面背景之顏色。

`pdfscreen` 巨集套件特別在畫面上占用一小區塊版面，其內提供數個按鍵，用以控制投影片之放映。譬如，點選向右之三角形按鍵可移入下一頁，向左之三角形可退回上一頁。引用 `pdfscreen` 時，若加入 `panelright` 選項，按鍵組小區塊版面將置於畫面右邊；反之，`panellleft` 選項則使按鍵組置於畫面左邊。因為此按鍵組所占的版面稍大，若簡報內容複雜，剩下之畫面可能不敷使用。本例子選擇不用內定之按鍵組版面，故加入 `nopanel` 選項。但是，畫面上仍須有控制按鍵；下一行之 `\bottombuttons` 指令即指示將一組簡單、不占空間的控制按鍵置於版面下方。接下來的兩行指令，`\screensize` 設定版面大小，`\margins` 設定四週留邊的大小。

最後，第 12 行指令 `\overlay{cwback.pdf}` 是一簡單的背景圖案。此一圖案是由 `cw \TeX` 所提供，但使用者可重新自行設計。若使用自行設計之背景圖案，圖檔請存為 PDF, JPEG, 或 PNG 格式。圖 15.9 是排版結果在 `Acrobat Reader` 上顯示之效果。以滑鼠點選畫面底下之選項白點，可跳至首頁 (`First`)，上一頁 (`Prev`)，下一頁 (`Next`) 等。

`pdfscreen` 巨集套件是以 `hyperref` 巨集套件為基礎，而後者提供網路文件超連結 (`hyperlink`) 之功能。舉例來說，文稿甲處有下列指令與文字：

```
台灣的\hyperlink{ltest}{ 經濟成長率}為 ...
```

其中，`ltest` 為自訂之標誌。文稿乙處出現經濟成長率資料處之前再鍵入指令 `\hypertarget{ltest}`，以 `pdf \LaTeX` 排版之後，甲處之“經濟成長率”標誌文字會以特別顏色排出。使用 `Acrobat Reader` 預覽時，以滑鼠點選“經濟成長率”，顯示畫面將跳至 `\hypertarget` 指令處。如果點選圖 15.9 畫面下端工作列上之 `Go Back`，即回到原畫面。



圖 15.9: PDF 投影片

15.5.2 特殊效果

圖 15.8 的例子使用 `pause` 巨集套件, 其功能是讓投影片的內容依序分段顯示。舉例來說, 若某投影片以 `itemize` 指令環境列示 3 點討論事項:

```
\usepackage{pause}
...
\begin{itemize}
\item 台灣的產出成長: 1895--1945 \pause
\item 由成長理論解釋台灣的產出成長 \pause
\item 台灣人與日本人之所得分配
\end{itemize}
```

我們在前兩項末端加上 `\pause` 指令。此一例子由 `pdf \TeX` 排版, 所產生之 PDF 檔案再經過後階段處理 (post-processing), 即可出現分項顯示之效果。具體言之, 在 Acrobat Reader 畫面上, 當放映此張投影片時, 一開始僅出現第一討論項。按下 `[Enter]` 鍵後, 畫面上將出現第 2 項, 再按 `[Enter]`, 全部三項才出現。

指令 `\pause` 可下於任何段落末端, 以產生上述之效果。不過, 以 `pdf \TeX` 排版後之 PDF 檔案須進一步處理才有效。後續處理之程式是由 PPower4 巨集套件所提供, 作者是 Klaus Guntermann。後續處理程式藉助 Java 軟體之功能。此軟體由 Sun Microsystem 發行, 可免費下載使用。若電腦中 PPower4 套件已安裝

完成, 點選光碟 \support\Java 檔案夾之 j2re-1_3_1_01-win-i.exe 即可安裝。

假設 Java 軟體與 MiKTeX 系統都安裝於 c: 硬碟, 且 PPower4 巨集套件已安裝完成, 則 c:\texmf\source\latex\ppower4 檔案夾內有 4 個 .jar 檔案。請先確認 c:\Program Files\JavaSoft\jre\1.3.1\lib 檔案夾內是否有以上 4 個檔案; 若沒有, 請複製進去。安裝完成之後, 利用 cwTeX 系統提供之 pp4.bat 批次檔即可作 PDF 檔之後階處理。此批次檔置於 c:\texmf\cwtex 檔案夾內。若 Java 程式安裝於 c: 以外之硬碟, 請修正其中之設定; 否則應保持原狀。

cwTeX 之例子檔案夾內有 foil-pp4.ctx 測試檔, 以 WinEdt 開啓, 先執行 cw-tex 轉換中文, 接下來以滑鼠點選 WinEdt 工作列上之 pdfTeX 快速鍵圖像, 將文稿排版成 PDF 檔案。之後, 進入 DOS 視窗, 執行下列指令進行後階段處理:

```
c:\xtemp>pp4 foil-pp4
```

完成之後, 以 Acrobat Reader 開啓 foil-pp4.pdf, 欣賞結果。

除了分段顯示之效果外, pdfscreen 與 PPower4 巨集套件還提供其他的顯示效果, 例如, 投影片畫面可由左至右逐漸打開, 也可以溶入、淡出。請參考兩巨集套件之說明。

15.5.3 引用 PDF 格式圖形檔

測試檔 foil-pp4.ctx 所排版之投影片含有一張圖形。請特別注意, pdfTeX 並無法處理 EPS 圖形檔, 僅能處理下列 3 種圖形格式檔: PDF, JPEG, 與 PNG。如果你原來有 EPS 格式之圖形檔, 可以先轉為 PDF 格式, 再以 \includegraphics 指令引入。

許多的畫圖軟體可以將圖檔轉換為 PDF 格式。譬如, CorelDraw 軟體提供“Publish to PDF”功能, PhotoShop 可以直接存成 PDF 格式。另外, Acrobat Distiller 軟體可以將 PostScript 檔案轉換成 PDF 格式。以上都是商業付費軟體, 如果沒有這些軟體, 可以使用 CTAN 上免費之 eps2pdf 軟體, 作者為 Nikola Jakšić。此一軟體可在 Windows 上執行, 它利用的是 Ghostscript 之功能。由 CTAN 下載, 或由光碟 \util 檔案夾取得軟體, 直接執行 eps2pdf.exe, 經過簡單的設定, 即可進行轉換動作。

第 11.2.2 節 (頁 231) 曾介紹如何使用 psfrag 巨集套件在 PostScript 圖形檔內加入中文字或數式。遺憾的是, 此一方法在 PDF 圖形檔上行不通。欲在 PDF 圖

形檔上加入中文, 可使用以下的間接方法。首先, 製作一 \LaTeX 檔案, 其內容除了以 `\includegraphics` 指令引入 PostScript 圖形, 並以 `\psfrag` 指令替換出中文字或數學式之外, 不含其他指令或文字。請特別注意, 全文設定區須加入

```
\pagestyle{empty}
```

否則版面下方會出現頁碼。假設檔名為 `fig.ctx`, 排版之後以 DVIPS 產生 `fig.ps`。接下來, 開啓 GSview 軟體讀入 `fig.ps`, 點選

```
File|PStoEPS
```

即轉換出 `fig.eps`, 此步驟的目的是找出圖形外框的大小。最後, 利用 `eps2pdf` 產生 `fig.pdf`, 此一含中文或數式之 PDF 檔即可引入投影片檔案內。引用 PDF 圖形檔仍然是使用 `\includegraphics` 指令, 但引用圖檔時, 不須加上延伸檔名, `pdf \LaTeX` 會自動搜尋以 `.pdf` 為延伸檔名之圖檔。

15.6 習題與解答

學校的教師經常出習題/考題給學生, 考試之後則提供解答。準備習題或考題時, 最好同時也備妥答案。Mike Piff 所寫的 `answers` 巨集套件即作此用途。從老師的角度來說, 習題/考題之解答最好是直接輸入題目之下, 排版時, 解答則另成一頁, 排版於習題後面。如果你是排版一本書, 習題可能散佈於書內各處, 解答則集中於書末。巨集套件 `answers` 的基本功能是将解答集中存入一檔案, 再於文稿末端, 或任何自行指定之地點引入。

圖 15.10 例子中, 第 1 行引入 `answers` 巨集套件; 第 2 行所引入之 `titlesec` 巨集套件與習題解答並無關係, 目的只是設定使用較小之 `sans serif` 字體。第 3 行指令如下:

```
\Newassociation{sol}{Solution}{ans}
```

第一選項創造一指令環境 `sol`, 文稿內之習題解答即輸入於此指令環境內。本例子有兩個題目, 解答即輸入於 `sol` 指令環境內。相對而言, `Solution` 也是一個新創造之指令環境, 其目的是用於設計習題頁每一習題之排版格式。如果接受內定之排版格式, 我們不須直接引用此一指令環境。

```

\usepackage{answers}
\usepackage[sf,small]{titlesec}
\Newassociation{sol}{Solution}{ans}
\begin{document}
\Opensolutionfile{ans}[ans1]
\section{\ctxfr  習題}
\fontsize{12}{16pt plus1pt minus.6pt}\selectfont
\begin{enumerate}
\item First exercise.
\begin{sol}
First solution.
\end{sol}
\item Second exercise.
\begin{sol}
Second solution.
\end{sol}
\end{enumerate}
\Closesolutionfile{ans}
\newpage
\section{\ctxfr  習題解答}
\input{ans1}
\end{document}

```

圖 15.10: 習題與解答 — 例1

圖 15.10 指令 `\Newassociation` 第3選項為 `ans`，這是 `answers` 自動創造之檔案，檔名為 `ans.tex`，其內即存放習題解答內容。請注意，以上三個選項名稱可自行選定。譬如，若第一選項名稱改為 `answer`，則文稿內輸入習題解答時，須置於 `answer` 指令環境內。

以上指令雖然選定檔案名稱 `ans.tex`，實際排版時須下一指令要求電腦開啓此檔案，以便開始接受輸入之解答。圖 15.10 第5行指令即作此用途。

```
\Opensolutionfile{ans}[ans1]
```

此行指令末端加上 `[ans1]` 選項，目的是把檔案名稱進一步改為 `ans1.tex`。若不加此選項，檔名即為內設之 `ans.tex`。檔名為何進一步改變？理由是在較複雜的文稿中，可能有必要開啓不同名稱的檔案以存放不同內容之解答。譬如，若是教科書單數題答案欲附於書後，雙數題答案打算排版於教師手冊內，則開始兩個不同名稱的檔案即可分別儲存單雙數習題解答。另外一種應用是假設一本書內有12章，每一章之解答可以存入各自之檔案內以利分別處理。相關之細節請參考

巨集套件內附說明檔。

如圖 15.10 倒數第 5 行所示, 解答輸入完成之後, 須加入下列指令:

```
\Closesolutionfile{ans}
```

以確認各習題已儲存於檔案內。倒數第 4 行 `\newpage` 讓版面前進到下一頁。下一行指令是以 `\section` 指令排版習題解答之標題; 實際上我們可以使用任何指令作任何設計。接下來以 `\input{ans1}` 指令將前面已儲存之習題解答引入文稿內; 最後一行指令結束文稿內容。

排版時, 解答將依原先題目之格式出現。圖 15.10 的例子是以 `enumerate` 指令環境排版習題, 解答也以同一格式出現。依 `answers` 巨集套件之設定, 解答編號將以粗黑體字排版。不過, 使用者可以重新設定。譬如, 在全文設定區加入下列一行指令:

```
\renewcommand{\solutionstyle}[1]{\fbox{#1.}}
```

解答編號將以正體字排版, 並加上一四方形, 如 $\boxed{1}$ 。巨集套件 `answers` 的原始設定就是使用 `\textbf{#1}`, 因此編號數字以粗體字排版。

圖 15.11 是第 2 個習題與解答的例子。本例中, 習題是排版於 `ex` 指令環境內, 此指令環境則是由 `\newtheorem` 所設定, 請參見第 9 章之說明。開啓解答檔案之指令為:

```
\Opensolutionfile{ans}[ans-ch\thechapter]
```

其中, `\thechapter` 是章編號之計數器。若本章為第 15 章, 則上列指令將開啓檔案 `ans-ch15.tex`。相對應的, 倒數第 2 行以 `\input` 指令引入本章之習題解答。因此, 若一本書有 16 章, 則上述之作法即可開啓 16 個習題檔案。

```
\documentclass[12pt]{book}
\usepackage{answers}
\Newassociation{sol}{Solution}{ans}
\newtheorem{ex}{Exercise}
\begin{document}
\Opensolutionfile{ans}[ans-ch\thechapter]
\section{Problems}
\begin{ex}
  First exercise ...
  \begin{sol}
    Solution to first exercise.
  \end{sol}
\end{ex}
\begin{ex}
  Second exercise ...
  \begin{sol}
    Solution to second exercise.
  \end{sol}
\end{ex}
\Closesolutionfile{ans}
\section{Solutions to Chapter \thechapter}
\input{ans-ch\thechapter}
\end{document}
```

圖 15.11: 習題與解答 — 例2

16 參考文獻

學術論文常引用參考文獻。第 7 章曾說明如何使用 `thebibliography` 指令環境排版參考文獻。首先，將所有引用的文獻輸入於指令環境內，每一篇文獻取一檢索名 (*key*)，文稿內再以 `\cite` 指令引用之。排版時， \LaTeX 自動替每一篇文獻加上編號，參見圖 7.2 (頁 106) 例子。

經常寫作論文的人都知道，多篇論文可能引用同一篇文獻。如果每篇論文都要鍵入此一文獻，不僅費時，也容易犯錯。為解決排版參考文獻的問題，Oren Patashnik 特別發展一套系統，稱為 $\text{Bi}\text{\TeX}$ ，其中包含了 `bibtex` 程式及一些設定參考文獻排版格式之檔案。 $\text{Bi}\text{\TeX}$ 排版參考文獻的方法如下：使用者先建立一參考文獻資料檔，每一項文獻取一檢索名稱。文稿內可以經由檢索名稱徵引文獻，排版時 $\text{Bi}\text{\TeX}$ 將由文獻資料檔內擷取所引用之文獻，排版於文稿末端。

$\text{Bi}\text{\TeX}$ 僅能處理英文或其他西洋文字。為了解決中文參考文獻，我們須加強 `bibtex` 程式之功能以處理中文，同時還須設計排版中文文獻之格式。新程式稱為 `cwbibtex`，這是翁鴻翎先生所寫。`cwbibtex` 系統雖然主要是用於處理中文稿，但也能處理純英文稿。

16.1 排版參考文獻之流程

在介紹排版參考文獻流程之前，我們先說明 $\text{Bi}\text{\TeX}$ 之原理。首先，使用者須把參考文獻鍵入於文獻資料庫檔案，此檔案須以 `.bib` 為延伸檔名。`\text{\LaTeX}` 例子檔案夾內所附之文獻資料檔例子名為 `test.bib`。如圖 16.1 所示，每一參考文獻須取一檢索名。第一篇文獻檢索名為 `HuangZG59`。論文內若欲徵引此一文獻，僅須下 `\cite{HuangZG59}` 指令即可。

參考文獻之排版牽涉到兩種格式，一個是論文末端排列參考文獻之格式，另一個是論文內引用參考文獻之格式。前一格式是利用文獻資料格式檔案 (`bibliography style`) 控制，其延伸檔名為 `.bst`。CTAN 內可找到各式各樣的格式檔

```

@Cbook{HuangZG59,
  author = "黃潤之",
  year = "1959",
  title = "臺灣之稻作農家經濟",
  address = "台北",
  publisher = "台灣銀行",
  series = "台灣研究叢刊第72種",
}

@Book{Davidson03,
  author = "Davidson, James W.",
  title = "The Island of Formosa, Past and Present",
  publisher = "Macmillan",
  address = "London",
  year = "1903",
}

@Article{WuTM97,
  author = "吳聰敏",
  title = "1945--1949 年國民政府對台灣的經濟政策",
  journal = "經濟論文叢刊",
  volume = "25",
  number = "4",
  pages = "521--54",
  year = "1997",
}

@Cmanual{SugarNote2,
  title = "第二次糖業記事",
  year = "1903",
  address = "台北",
  organization = "臨時臺灣糖務局",
  key = "《第二次糖業記事》",
}

```

圖 16.1: 文獻資料庫

案, 適用於各種類型的期刊。但這些格式都僅能處理英文或其他西方國家之文字。就中文稿為言, 我們須另行設計排版中文文獻之格式。cw_TE_X 所提供的格式檔例子名為 `cw.bst`, 它可以同時處理中英文文獻。如果你不滿意其中之格式, 後文將簡單介紹如何自行製作 `.bst` 格式檔。

論文內徵引文獻之方法也有多種格式, 有些是以文獻之編號徵引, \LaTeX 內定

之格式就是以文獻之編號徵引，請見圖 7.2 (頁 106) 之例子。有些則以作者名字加上出版日期徵引，例如，“根據矢內原忠雄 (1937) 之論述...”。若欲使用 L^AT_EX 後一格式，我們須引用巨集套件，例如 `harvard` 或 `natbib` 巨集套件。

綜合以上所述，若文獻資料庫檔名為 `test.bib`，文獻資料之排版格式為 `cw.bst`，而徵引參考文獻之格式擬使用 `natbib` 巨集套件，則排版時所需之相關指令如下：

```
\usepackage{natbib}
\renewcommand{\refname}{ 參考文獻}
...
\begin{document}
...
\bibliographystyle{cw}
\bibliography{test.bib}
\end{document}
```

以上第 1 行指令引用 `natbib` 巨集套件，16.4 節 (頁 318) 將介紹使用方法。文稿以 `article` 文件類別排版時，參考文獻之前會自動加入英文標題 **References**。第 2 行指令將之改為「參考文獻」四字。若文稿是以 `book` 文件類別排版，英文標題為 **Bibliography**，更改指令為：

```
\renewcommand{\bibname}{ 參考文獻}
```

本例中，倒數第 3 行指令選用 `cw.bst` 之文獻格式，倒數第 2 行指令則指定文獻資料檔 `test.bib`。若文獻資料檔名為 `mybib.bib`，則倒數第 2 行應改為：

```
\bibliography{mybib.bib}
```

下一節將進一步說明。

第 2 章的 `examp6.ctx` 為排版參考文獻之例子。欲排版此一例子，須依序執行以下 5 道指令：

```
c:\texmf\cwtex\examples>cwtex -d=c:\xtemp examp6
c:\xtemp>latex examp6
c:\xtemp>cwibibtex examp6
c:\xtemp>latex examp6
c:\xtemp>latex examp6
```

以上之每一步驟都可以滑鼠點選 WinEdt 工作列上之快速鍵圖像直接執行。其

中, `cwibibtex` 圖像是一本攤開的書上有一個大問號, 請見圖 3.3 (頁 43)。

以 `latex` 排版之後, 文稿所引用的參考文獻之檢索名全部錄於 `examp6.aux` 輔助檔案內。第 3 行指令之 `cwibibtex` 程式會自文獻資料庫檔案 `test.bib` 內取出論文所引用之文獻, 經過排序處理之後, 存放於另一輔助檔案 `examp6.bbl` 內。以文字編輯軟體檢視此檔案內容, 我們發現檔案前端是中文字體及巨集指令之定義, 後半段則是 `thebibliography` 指令環境, 其內容類似圖 7.2 (頁 106) 之例子。由此可知, `cwibibtex` 程式之作用是在文稿末端加入一個 `thebibliography` 指令環境。此外, `examp6.blg` 檔案記錄排版參考文獻過程之訊息; 任何錯誤或警告訊息都記錄於其內。

根據以上說明, 欲排版徵引文獻須執行 5 道指令。若文稿反覆修改, 每次排版都須執行那麼多道指令, 還是很不方便。不過, 如果文稿修改過程中, 並未添加新的參考文獻, 則重新排版時僅須執行前兩道指令即可, 後面 3 行指令是不需要的。

16.2 調整設定

依上一節所述, 排版參考文獻的第 3 步驟是啟動 `cwibibtex` 程式, 此程式為一批次檔, 內容如下:

```
copy %1.ctx c:\xtemp
copy c:\texmf\cwtex\examples\test.bib c:\xtemp
rem copy c:\tex\bibtex\mybib.bib c:\xtemp
c:
cd \xtemp
cwibibtex.exe %1
copy %1.bbl bak.bbl
copy cinput.tex+bak.bbl %1.bbl
```

處理文獻資料時, `cwibibtex` 須讀取文稿原檔, 若檔名為 `paper.ctx`, 第 1 行指令將 `paper.ctx` 檔案複製至工作檔案夾。第 2 行指令則是將文獻資料檔 `test.bib` 複製於工作檔案夾內。若你個人建立之文獻資料檔取名為 `mybib.bib`, 且置於 `c:\tex\bibtex` 檔案夾內, 請將第 2 行刪除, 第 3 行則改為:

```
copy c:\tex\bibtex\mybib.bib c:\xtemp
```

若個人文獻資料檔是置放於 `d:\tex\bibtex`, 第 3 行請改為:

```
copy d:\tex\bibtex\mybib.bib c:\xtemp
```

參考文獻通常是排版於文稿末端，各篇文獻會以指定之方式排序。若文稿內同時引用中、英文獻，中文文獻將排於英文之前。如果你希望英文文獻排在前面，則以上第 6 行指令請加入 `-e` 選項：

```
cwbibtex.exe -e %1
```

排版於文稿末端之文獻都已經過排序。目前，`cwbibtex` 提供兩種中文排序方法，第 1 種（內設）是先筆劃後部首，第 2 種是按 `big-5` 之順序。如果要使用第 2 種排序法，請加入選項：

```
cwbibtex.exe -s2 %1
```

16.3 建立文獻資料檔

`cwTeX` 所提供的文獻資料檔例子 `test.bib` 置於 `c:\texmf\tex\bibtex` 檔案夾內，可供參考。以文字編輯軟體開啓此檔，其內容如圖 16.1 (頁 312) 所示之 4 篇文獻。仿照其內的例子即可加入自己所需要之文獻，

依 `BiBTeX` 之設計，參考文獻種類有 13 種，包括論文 **article**，書籍 **book**，碩士論文 **masterthesis**，技術報告 **techreport** 等。建立文獻資料庫時，每一文獻種類內有一些必須填入之資料格位 (field)，稱為 `required fields`；有一些是可填可不填，稱為 `optional fields`。輸入資料時，若某項 `required fields` 漏過未填入，執行程式時會出現警告訊息；但 `optional fields` 則無所謂。以 **article** 為例，`required fields` 為 `author`, `title`, `journal`, `year` 等 4 項；`optional fields` 則包括 `volume`, `number`, ..., `note` 等 5 項。

實際排版時，`cwbibtex` 程式會取用各格位內之資訊，並依照文獻格式檔 `.bst` 所指定之格式排版每一文獻。網路上可找到許多專家們所設計之格式檔案。這些格式之設計是以英文文獻為對象，使用它們排版中文文獻並不適合。為了能同時處理中英文參考文獻，我們特別設計一 `cw.bst` 格式。若欲使用此一格式，則建立文獻資料檔時英文論文應使用 **article** 種類；中文論文則使用 **carticle** 種類。同理，英文書籍以 **book** 格式，中文書籍則使用 **cbook**；餘此類推。請參考前面圖 16.1 之例子。

底下列出文獻種類之名稱及其格位。每一類別都列出兩個名稱,前者用於排版英文文獻,如 **article**,後者用於排版中文文獻,如 **carticle**。

article/carticle 期刊/雜誌之文章。

Required fields: author , title , journal , year .

Optional fields: volume , number , pages , month , note .

book/cbook 有出版者之書籍。

Required fields: author or editor , title , publisher , year .

Optional fields: volume or number , series , address , edition , month , note .

booklet/cbooklet 無出版者或出版機構之書籍。

Required field: title .

Optional fields: author , howpublished , address , month , year , note .

conference/cconference 同 **Inproceedings**。

inbook/cinbook 某書的一部份,可能是某一章節或特定頁碼。

Required fields: author or editor , title , chapter and/or pages , publisher , year .

Optional fields: volume or number , series , type , address , edition , month , note .

incollection/cinollection 某書的一部分,有特定標題。

Required fields: author , title , booktitle , publisher , year .

Optional fields: editor , volume or number , series , type , chapter , pages , address , edition , month , note .

inproceedings/cinproceedings 會議論文集文章。

Required fields: author , title , booktitle , year .

Optional fields: editor , volume or number , series , pages , address , month , organization , publisher , note .

manual/cmanual 技術文件 (Technical documentation)。

Required field: title .

Optional fields: author , organization , address , edition , month , year , note .

mastersthesis/cmastersthesis 碩士論文。

Required fields: author , title , school , year .

Optional fields: type , address , month , note .

misc/cmisc 無法歸類之文件或書籍。

Required fields: none.

Optional fields: author , title , howpublished , month , year , note .

phdthesis/cphdthesis 博士論文。

Required fields: author , title , school , year .

Optional fields: type , address , month , note .

proceedings/cproceedings 會議論文集。

Required fields: title , year .

Optional fields: editor , volume or number , series , address , month , organization , publisher , note .

techreport/ctechreport 學校或特定機構發行之報告, 通常冠以系列名稱。

Required fields: author , title , institution , year .

Optional fields: type , number , address , month , note .

unpublished/cunpublished 標示有標題與作者之文稿, 但未正式出版。

Required fields: author , title , note .

Optional fields: month , year .

以文字編輯軟體檢視 test.bib , 即不難了解文獻資料檔案之輸入方法。以下僅說明幾個要點。

1. 每一份參考文獻含數個格位之資料。輸入資料時, 請特別注意, “%” 並無 \LaTeX 註銷指令之功能。
2. 英文之姓名可輸入為 "Myers, Wykeham W." , 或者 "Wykeham W. Myers" 。兩位以上之作者則以 “and” 隔開, 例如: "Maddison, Angus and Harry van Ooststroom" 。
3. 作者名 Robert E. Lucas, Jr. 應輸入為 "Lucas, Jr., Robert E." 。
4. 輸入特別之字母或符號時, 前後請加上大括號, 例如:

```
author = "Kurt G{\{o\}}del",
```
5. bibtex 會嘗試區別姓與名, 但其判斷有時難免出現錯誤, 特別是當兩位以上之中文作者名字以空白隔開時。如果你要求以特定之順序排版姓名, 可將姓名置於左右大括號內。
6. test.bib 之格式是將兩位以上之中文作者名字以一小黑點隔開, 輸入之指令為 "謝森中。潘盛藩"。
7. 輸入英文 title 時, 除了介系詞之外, 每一個單字第一個字母應大寫。
8. 中英文 title 末端請勿加句點。標點符號之格式由 .bst 格式檔控制。

進一步的說明請見 btxdoc 檔案。

16.4 natbib 巨集套件

每一種期刊各有其參考文獻之排版格式，而論文內引用參考文獻之格式也不同。舉例來說， \TeX 內定之格式是將每一篇文獻編上號碼，論文內直接引用編號。譬如，若某文獻編號為 7，排版之後論文內引用文獻之前後文為“根據 [7] 之分析...”，請見圖 7.2 (頁 106) 例子。

另一種常見徵引文獻之格式稱為“author-date”，亦即作者姓名加上論文出版日期。例如，若作者姓名為 Lucas，論文發表於 1983 年，則引用文句將為“根據 Lucas (1983) 之分析...”，或者“經濟成長率的高低與儲蓄率並無相關 (Lucas, 1983)。”若使用 author-date 格式，論文末端之參考文獻不加上編號，其排列通常是依作者名字之字母順序。

上面已經說明，文稿末端參考文獻之格式是以 bibliography style 格式控制，例如 cw.bst。但是，論文內引用文獻之格式是以巨集指令控制；其中，natbib 巨集套件是廣為使用的一套，作者為 Patrick W. Daly。natbib 巨集套件主要用於設定 author-date 格式，但也可以用於 \TeX 原始之編號格式。以下之僅簡單介紹 author-date 格式。

16.4.1 徵引文獻之指令

\TeX 原有徵引文獻之指令為 `\cite`。為了處理“author-date”之格式，natbib 巨集套件將之延伸為 `\citet` 與 `\citep`，前者代表 *textual citations*，主要是將文獻作者當作主詞時使用；後者代表 *parenthetical citations*，主要是將文獻作為行文末端之補充時使用。

圖 16.2 例示指令使用方法及排版結果，請特別注意第 1 行與第 3 行之差別。第 1 行徵引中文獻，`\citet` 指令緊接其前之中文字。反之，第 3 行徵引英文文獻，指令與其前之中文字之間須留出一個空隔。第 2 行與第 7 行之例字說明，指令之前可加入選項，以排版頁碼或其他文字。

在 author-date 的格式之下，每一篇文稿必須有作者或編者名字。但是，有一些出版品僅列出出版機構名稱，並沒有作者或編者名字，政府機構出版品多屬此類。我們特別透過在 `manual` 與 `cmanual` 文獻類別解決此一問題。圖 16.1 的例子中，《第二次糖業記事》歸類為 `cmanual`，出版機構為「臨時臺灣糖務局」，但無作者。請注意，輸入時須加入“key”一項，它有兩個作用。第一，文稿末端排版參

依據吳聰敏 (1997) 之 ...	依據 <code>\citet{WuTM97}</code> 之 ...
依吳聰敏 (1997, 頁 30) ...	依 <code>\citet[頁 30]{WuTM97}</code> ...
依 Davidson (1903) 之 ...	依 <code>\citet{Davidson03}</code> 之 ...
依據吳聰敏之研究 ...	依據 <code>\citeauthor{WuTM97}</code> 之研究 ...
依《第二次糖業記事》(1903) ...	依 <code>\citet{SugarNote2}</code> ...
... 結論 (吳聰敏, 1997)。	... 結論 <code>\citep{WuTM97}</code> 。
... 結論 (吳聰敏, 1997, 頁 30)。	... 結論 <code>\citep[頁 30]{WuTM97}</code> 。

圖 16.2: natbib 巨集套件徵引文獻指令

考文獻時，將以“key”之內容排序。第二，文稿徵引文獻時，原應出現作者名字部分，將以“key”之內容替代。請見圖 16.2 之結果。

如果是無作者名字之英文出版品，請利用 **manual** 文獻類別。例如，若書名為 *A System of National Accounts*，出版於 1993 年，檢索名設為 SNA93。以 **manual** 類別輸入於文獻資料庫時，“key”一項應與書名完全相同。文稿內引用此一文獻時，若直接使用 `\citet{SNA93}`，排版結果將如下例所示：

根據 A System of National Accounts (1993), ...

但英文排版規範是，書名應使用斜體字。故較好的輸入方法是利用 **natbib** 巨集套件所提供的 `\citeauthor`：

根據 `\textit{\citeauthor{A System of National Accounts}}`
(1993), ...

排版之後，書名將為斜體字。

某些文獻在文稿內直接輸入可能反而較方便。以上面例子來說，我們可直接輸入：

根據 `\textit{A System of National Accounts}` (1993), ...

而不須使用 `\cite{SNA93}` 指令。但此一作法的問題是，此項文獻不會自動列入文末參考文獻內。為解決此問題，我們可使用 `\nocite{SNA93}` 指令將該文獻納入文稿末之參考文獻。若文稿內輸入 `\nocite{*}` 指令，則文獻資料檔內全部的文獻都將排版於文稿末端，這有助於檢視文獻資料檔內全部的文獻內容。

徵引之文獻經過排序後，將排版於文稿末端，其格式為 hanging indent，字體與正文相同。使用可以透過指令改變排版格式。譬如，內定之文獻間距比正常行距約加大一倍。欲自行設定間距為正常行距加大 2pt，可在全文設定區加入下列指令：

```
\setlength{\bibsep}{2pt}
```

如果要設定 hanging indent 之內縮距離為 1 公分，指令為：

```
\setlength{\bibhang}{1cm}
```

巨集套件 natbib 提供許多控制指令，欲了解詳情，請自行參閱其說明檔。

16.4.2 流通文稿

若論文是使用 BiB_TE_X 排版參考文獻，則傳送最後之完稿給排版公司時，須把文獻資料檔及格式檔一併送出，否則排版者無法完成排版工作。但是，若排版者並未使用 BiB_TE_X 系統，我們必須把所引用之文獻直接添加於文稿末端。

前面已說明，排版過程中會產生一輔助檔案 `examp6.bbl`，其內容即為論文所徵引之有文獻，但所有之中文字已轉換為 T_EX 指令。執行下列指令：

```
c:\xtemp>tex2xtc examp6.bbl
```

可將檔案轉換成 `examp6.xtc`，其內含正常中文字。檔案內可找到一行指令：

```
\begin{thebibliography}{15}
```

此行開始到檔案末端定義 `thebibliography` 指令環境，請移入排版文稿末參考文獻應出現之位置。文稿原設定參考文獻格式及資料檔之兩行指令則註銷，或直接刪除。

此外，`examp6.xtc` 檔案前端之指令分兩部分，第一部分是中文字體之定義指令，大都是以 `\providecommand` 指令開頭。其後，以 `\newcommand` 所定義者為排版參考文獻所需之指令，這一部分的指令（一直到 `thebibliography` 指令環境之前），請移入排版文稿之全文設定區。經此整理，文稿檔案本身已納入所有引用之參考文獻項目，直接以 latex 編譯即可排版出參考文獻。

16.5 製作文獻格式檔

若你排版的是英文文稿，則 CTAN 上有許多各式各樣的 .bst 檔案供下載使用。不過，英文格式不見得適用於中文文獻。譬如，英文書籍常以斜體字排出，*A Theory of National Income*，但中文書可能排為《國民所得理論》。又如，若某篇論文載於某人所編之論文集內，排版形式可能是 “in *A Theory of National Income*, ed. by ...”；而中文形式可能是：“刊於《國民所得理論》，... (編)”。為了解決中英文獻並列的問題，格式檔案內除了 **book** 文獻種類外，必須有一對應的 **cbook** 文獻種類以編排中文書籍。

如何製作一中英文適用之 .bst 檔案？方法之一是選用一英文 .bst 檔，修改其內之格式以排版中文文獻。欲修改 .bst 檔案，你必須了解 **bibtex** 程式之語法。有興趣者，請參考 **btzhak** 說明檔。

你也可以製作全新格式的英文 .bst 檔案。欲製作新文獻格式檔案，最方便的方法是利用 **makebst** 巨集套件，作者為 Patrick W. Daly。我們所製作的 **cw.bst** 中文格式檔，即先以 **makebst** 產生英文格式檔，再進一步修改以處理中文文獻。

16.6 listbib 巨集套件

文獻資料庫之內容會隨著時間而越來越龐大。為了方便參考使用，我們有時候必須將其內容列印出來。欲列印文獻資料庫內容，可使用 **listbib** 巨集套件，作者為 Volker Kuhlman。

使用方法很簡單，假設文獻資料庫檔名為 **test.bib**，則以文字編輯軟體鍵入底下內容：

```
\documentclass{article}
\usepackage{listbib}
\begin{document}
\bibliography{test}
\end{document}
```

必要時，還可加入其他排版指令，例如選定英文字體大小。假設以上檔案取名為 **testlist.ctx**，請將之複製於工作檔案夾內，**test.bib** 也須複製於同一檔案夾內，執行以下指令：

```
cwtx testlist  
latex testlist  
cwbibtex testlist  
latex testlist
```

即得到一版面清爽的文獻資料內容。

17 版面設計

排版書籍或長篇文稿時，版面設計是重點之一。所謂版面設計，我們指的是版面尺寸、章節標題、頁眉與頁足。L^AT_EX 對於以上幾點都有內定之設定，但修改彈性很小。若某書是以 L^AT_EX 標準的版面格式排版，我們很容易一眼就看出來。

要改變版面設計，最好的方法是使用現成的巨集套件。本章將介紹三套巨集套件。第一套是 `geometry`，用於設定版面尺寸。第二套是 Javier Bezos 所寫的 `titlesec`，用於設計章節標題。此一巨集套件提供簡易與進階兩種指令方式以變更章節標題，7.5.3 節（頁 117）介紹的是簡易指令。如果你只是要改變標題字體與大小，或者只是要改變標題之排版位置，簡易指令即可滿足需求。如果要進一步更改標題設計，則須使用進階指令。

最後，我們將介紹排版頁眉與頁足的 `fancyhdr` 巨集套件。利用以上的巨集套件，可設計出令人滿意的版面出來。下一章所將介紹的書籍排版例子，即大量運用本章之指令。

17.1 `geometry` 巨集套件

上一小節所介紹的指令可用以控制版面大小與位置，但使用上仍嫌複雜。本小節將介紹巨集套件 `geometry`，使用上較方便，作者為 Hideo Umeki。介紹指令之前，先簡單說明幾個概念。從紙面列印的角度來看，一張白紙上包含可列印區域（printable area）及四周邊緣（margins）；可列印區域（又稱為 total-body）進一步區分為正文方塊（text area），頁眉，頁足，與邊註（marginal notes）；而四周邊緣可分上下左右四部分。

如圖 17.1 所示，設定正文方塊大小之指令是 `textwidth` 與 `textheight`。請注意，在 `geometry` 巨集指令內，設定指令之前端並無反斜線。以上兩道設定指令對應 L^AT_EX 之 `\textwidth` 與 `\textheight`。相對於 `textwidth` 指令，`width` 指令為 `textwidth` 加上邊註（marginal notes）之寬度。

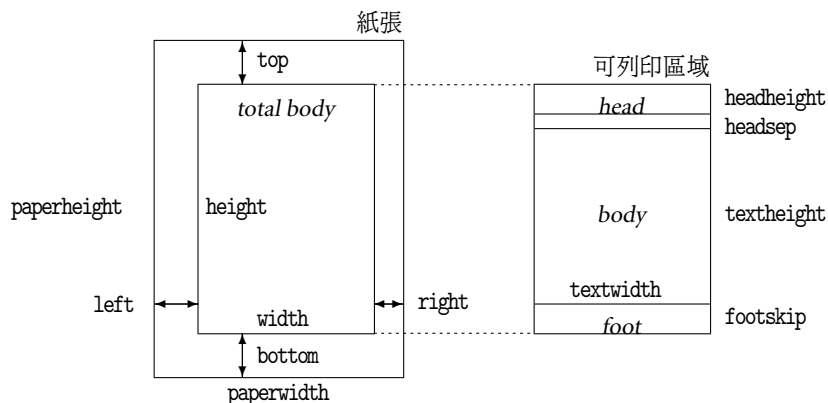


圖 17.1: geometry 巨集套件設定指令圖示

說明: 此圖取自 geometry 巨集套件說明檔。

根據圖 17.1, 若要以 geometry 巨集套件指令設定正文方格區域之尺寸, 僅須在全文設定區加入下列一行指令:

```
\usepackage[textwidth=5in,textheight=8in]{geometry}
```

如果要進一步設定左邊緣為 1in, 上邊緣為 1.5in, 設定指令為:

```
\usepackage[textwidth=5in,textheight=8in,
left=1in,top=1.5in,nohead]{geometry}
```

請注意, geometry 巨集套件已設計讓指令可直接拆為兩行以上, 前面各行末端不須加入註銷指令 %。本例中還加入 nohead 設定, 這等於是把 headheight 與 headsep 都設為零。

由圖 17.1 可知, 以上之設定並不完整。譬如, 我們至少還須加入 paperheight 與 paperwidth 指令, 設定紙張大小。不過, 若使用者不設定, geometry 會取用內定值。例如, 若紙張種類不設定, 巨集指令即取用 letterpaper, 並進一步決定 paperheight 與 paperwidth 之值。如果是使用 A4 尺寸紙張, 應加入 a4paper 設定項; 或者也可以直接加入於 \documentclass 選項內。

除了圖 17.1 所介紹的設定指令外, 還有許多其他指令可供使用, 以下僅介紹其中較常用者。欲了解所有細節, 請參考說明檔。假設選用 A4 紙張, 可列印區域

(正文方塊加上頁眉與頁足)之高度設為9in, 左右邊緣又留1in 空白, 指令為:

```
\usepackage[a4paper,height=9in,hmargin={1in,1in}]{geometry}
```

其中, `hmargin={1in,1in}` 是用於設定左右邊緣之空白。以上之設定方法是直接將各設定項加入於 `\usepackage` 指令內, 另一種方法是將所有之設定另外輸入於 `\geometry` 指令內, 如下例所示:

```
\usepackage{geometry}
\geometry{a4paper,height=9in,hmargin{1in,1in}}
```

指令效果完全相同。本例中, 長度直接設定為英吋單位, 但我們也可以下列方式設定:

```
\usepackage[hmargin{0.1\textwidth,0.1\textwidth}]{geometry}
```

以下列出一些較常用的設定指令:

- `nohead` 等於 `headheight=0pt,headsep=0pt` 。
- `nofoot` 等於 `footskip=0pt` 。
- `hscale` 可列印區域之寬度與紙張寬度之比率, 例如 `hscale=0.8` 。
- `marginparwidth` 邊註 (marginal notes) 之寬度。
- `marginparsep` 正文方塊與邊註之間距。
- `width` 等於 `textwidth + marginparsep + marginparwidth` 。
- `twoside` 奇數頁 (右頁) 之列印區域會稍靠左, 雙數頁 (左頁) 之列印區域則稍靠右; 參見下一指令。
- `twosideshift` 以 `twoside` 方式列印時, 可用此一指令設定奇數頁左移 (與雙數頁右移) 之距離。若不自行設定, 內設值為 `20pt`。譬如, 若奇數頁要左移0.5公分, 指令為: `twosideshift=-.5cm` 。
- `hoffset` 同 \TeX 之 `\hoffset` 指令。
- `voffset` 同 \TeX 之 `\voffset` 指令。

本書所使用之指令如下:

```
\geometry{paperheight=23.5cm,paperwidth=17cm,textwidth=11.5cm,
  textheight=18.5cm,left=2.75cm,twosideshift=-.75cm,top=.5cm,
  bottom=1.25cm,footskip=1.75cm,marginparwidth=2cm,twoside}
```

其中, `twosideshift` 是用以調整單雙頁之版面位置。除了以上指令之外, 我們還另外使用 `\hoffset` 與 `\voffset` 指令控制列印位置。

17.2 titlesec 巨集套件進階指令

欲排版章節標題, 可使用 `titlesec` 巨集套件。此一套件之功能甚佳, 但指令複雜。[7.5.3](#) 節會介紹如何使用其簡易指令以控制標題之排版, 本節將介紹進階指令。進階指令主要有 `\titleformat` 與 `\titlespacing` 兩項, 前者設計章節標題式樣, 後者控制間距。此外, 還有一些配合使用之控制指令。底下將以幾個例子說明使用方法, 這些例子係參考巨集套件之說明檔而來。

使用 `titlesec` 巨集套件時, 我們仍使用 \LaTeX 之 `\chapter` 與 `\section` 等指令排版章節標題, 但其格式則以 `titlesec` 提供之指令設計。為求文稿風格統一, 各章節標題應採用相同設計, 因此, 改變章節標題之指令通常是放在全文設定區。不過, 必要時 `\titleformat` 指令也可以用在特定章節標題之前。章節標題分兩部分: 第一部分為章節編號, 第二部分為標題文字。因此, `\titleformat` 指令一部分用於控制編號, 一部分用於控制標題文字。

假設我們以 `book` 文件類別排版, 每一章之標題要居中編排, 章編號希望以「第2章」格式出現; 節與小節之標題仍沿用原有格式, 但使用 `sans serif` 字體。下列指令可設計出以上之標題:

```
\usepackage[sf]{titlesec}
\ctxfdef{\chapter}{\ctxfm}{\ctxfr}
\titleformat{\chapter}[display]{\centering\LARGE\sf}
  {\第\ \thechapter\ 章}{0.2cm}{}
```

第1行為 `titlesec` 巨集套件之簡易指令, 除了選用 `sans serif` 字體外, 其餘皆依 \LaTeX 原有格式。第2行指令設定標題使用圓體字。第3-4行是 `\titleformat` 設計指令, 因為選項太長, 故拆為兩行。請注意, 依 \LaTeX 之規範, 指令拆為兩行時, 第1行末端應加上註銷符號 `%`。不過, `titlesec` 巨集指令之語法特別, 註銷符號 `%` 可省略不加。

巨集套件 `titlesec` 之指令相當複雜，為便於參考，我們將較重要者列於表 17.1。各選項之意義簡單說明如下。

- 第 1 選項 `command` 選擇欲更改設計之標題。本例選用 `\chapter`，表示欲更改章之標題設計。如果要更改節標題，應填入 `\section`，餘此類推。
- 第 2 選項 `shape` 設定章節編號與標題文字之排版方式。本例中使用 `display` 選項，這是 \TeX 原來排版章標題之格式：編號與標題文字分兩行排出。除了 `display` 之外，其他可能的設定包括 `runin`，`hang` 等，以下將進一步介紹。此選項若空白，內設值為 `hang`，這是 \TeX 排版節標題之格式，亦即標題文字接於編號之後，排於同一行。
- 第 3 選項 `format` 可填入排版指令，此處之排版指令對於章節編號與標題文字都有作用。本例使用 3 個指令，`\centering` 使標題與編號居中編排，`\LARGE` 選用較大字體，`\sf` 則選用 `sans serif` 字體。
- 第 4 選項 `label` 用於控制章節編號之排版。請注意，我們並未加入中文字體指令，欲選用中文圓體字，請在全文設定區加入以下指令：

```
\ctxfdef{\chapter}{\ctxfm}{\ctxfr}
```

本例之標題編號採用為「第 x 章」格式。故排版第 2 章時，標題編號將為為「第 2 章」，字體大小上一選項所設定之 `\LARGE`。

- 第 5 選項 `sep` 控制編號與標題文字之間距。若第 2 選項選用 `display` 格式，此間距設定上下兩行之間距；若是 `hang` 或者 `runin` 格式，編號與標題文字排於同一行，此選項即設定水平間距。
- 第 6 選項 `before` 是用於控制標題文字之排版。換言之，此部分之指令僅施用於標題文字，編號不受影響。此選項可能含一個或多個指令，最後一個指令可帶有一個參數 (`parameter`)，此參數即代表標題文字。底下將以例子說明使用方法。
- 最後一個選項 `after` 之內容將附加於標題文字後面。譬如，要在章標題文字之後加上一短線，可利用此一選項為之。本例中，6-7 選項內容皆為空白。排版結果每一章之標題將居中，編號與標題文字之間距為 0.2cm。

本例之章節編號以阿拉伯數字編排，有人偏好以國字數字編排。本例中，若第 4 選項之內容改為：

表 17.1: titlesec 巨集套件進階指令

`\titleformat{ command}[shape]{format}{label}{sep}{before}{after}`

- *command*: 欲重新設計之指令, 如 `\chapter` , `\section` , `\subsection` 等。
- *shape*: 設定標題格式。可用選項如下: `hang` , `block` , `display` , `leftmargin` , `rightmargin` , `runin` , `drop` , `wrap` , `frame` 。
- *format*: 此項內之指令將同時控制章節編號與標題文字之排版方式。
- *label*: `label` 是指章節編號及附加符號。此項內之指令是用以控制 `label` 之排版。
- *sep*: 控制章節編號與文字標題之間距。
- *before*: 用於控制章節標題之排版。此選項可能含一個或多個指令, 最後一個指令可帶有一個參數, 此參數即代表標題文字。
- *after*: 附加於標題文字後之控制指令。

`\titlespacing{ command}{left}{beforesep}{aftersep}{right}`

- *command*: 欲重新設計之指令, 如 `\chapter` , `\section` 等。
- *left*: 章節標題「左邊」之空白, 因 *shape* 選項之不同而異。若選用 `runin` , 此項設定是指內縮 (`indent`) 之大小。
- *beforesep*: 排版章節標題之前所留之空白。
- *afterskip*: 章節標題之後所留之空白。
- *right*: 若 *shape* 選項是選用 `hang` , `display` , `block` 或 `display` 之一時, 可進一步加此選項於標題末端。

控制指令

- `\titlerule[height]`: 畫橫跨正文寬度之線條; *height* 設定粗細, 若不設定, 內定值為 0.4pt。此指令通常置於 *shape* 選項內。
- `\titleline[height]{text}`: 同上一指令; 但線條替代以自行選定 *text*, 如細點。

說明: `\titlespacing` 指令若採 * 號形式, 變成 `\titlespacing*` , 則標題後之段落不內縮 (`indent`)。反之, 若指令不加 * 號, 段落首行文字將內縮。

{第一章}

則標題文字中之編號將改為國字數字。但因為國字之數字編號不會自動跳加, 第 2 章之標題編號仍然排為「第一章」。如果不怕麻煩, 一個解決的方法是在每一章 `\chapter` 指令之前重新定義 `\titleformat` 指令, 即可排出正確之國字編號。

除了每章之前下指令的麻煩之外, 以國字編號還有一個困難。中文數字不容易表現章節層級, 若一本書區分章、節、小節, 以國字編章節號碼時, 有人用「第參章第五節」, 有人則用「第三章第五節」, 章節之層級不容易清楚分辨。

17.2.1 彩色或灰階色標題文字

上面例子中, `\titleformat` 指令第 6 選項空白。此選項主要是用於排版標題文字。在 \TeX 中, 我們可以使用 `color` 巨集套件之 `\textcolor` 指令將文字段落變成灰階色。譬如, 先定義 `light` 為灰階度 0.60 之灰階色:

```
\usepackage{color}
\definecolor{light}{0.60}
```

之後, 以下指令即可將「章節標題」四個字變成灰階色。

```
\textcolor{light}{  章節標題}
```

色彩與灰階色之定義方法請參見第 12 章之說明。

不過, 以上之指令事實上並未完全解決問題, 原因是若使用 `\textcolor` 指令, 各章節的標題文字須直接置於 `\titleformat` 指令的大括號內; 但是每一章節之標題文字並不相同。`titlesec` 巨集套件提供的解決辦法如下: 定義一新巨集指令, 其內可自動填入章節標題文字, 再將此巨集指令引入 `\titleformat` 指令選項內。實際作法如下: 首先, 在全文設定區定義指令 `\mytitle` :

```
\newcommand{\mytitle}[1]{\textcolor{light}{#1}}
```

接下來再將 `\mytitle` 指令填入 `\titleformat` 指令第 6 選項內。再以上的例子來說明, `\titleformat` 指令將變成:

```
\titleformat{\chapter}[display]{\centering\LARGE\sf}
{第\ thechapter\  章}{0.2cm}{\mytitle}
```

排版時，每一章之標題文字將自動代入，以灰階色排版。

圖 10.10 (頁 197) 的例子曾介紹一巨集指令 `\cspp`，其功能是将中文字間距拉大為選定之距離。利用此一巨集指令，我們可將章節標題之中文字距拉大。欲創造此一效果，首先定義 `\cspp` 巨集指令，並將 `\mytitle` 修改如下：

```
\newcommand{\cspp}[1]{\let\zori=z
\def\zx{\hskip 2mm plus0.2pt minus0.1pt}
\let\z=\zx \mbox{#1}\let\z=\zori}
\newcommand{\mytitle}[1]{\textcolor{light}{\cspp{#1}}}
```

再與上述之 `\titleformat` 指令合併使用即可。

以上指令僅將章標題文字轉為灰階色，但章編號部分並不受影響。如果章編號也要以同一灰階色排版，必須另作處理。`\titleformat` 排版章節編號之指令是置於第 4 選項 *label* 內。因此，若將第 4 選項之內容改為：

```
\textcolor{light}{第\thechapter\章}
```

即可得到滿意的結果。

17.2.2 設計章節排版例子

在標準的 \LaTeX 格式中，章編號與標題文字是分兩行排版，此一格式稱為 *display*。下一個例子則採用 *hang* 格式，編號與標題文字將排於同一行。這是 \LaTeX 排版節標題之方式，指令如下：

```
\titleformat{\chapter}[hang]{\hspace*{.2\textwidth}%
\fontsize{45}{20pt}\selectfont\bf}
{\thechapter}{4mm}{\mytitle}
```

第 3 選項排版章編號時，先右移 0.2\textwidth 距離，再選用 45pt 之粗體字。第 4 選項內僅有 `\thechapter` 一項指令，表示除了編號之外，不加入其他文字。第 5 選項將編號與標題文字之間距控制為 4mm。排版標題文字之指令如下：

```
\newcommand{\mytitle}[1]{\raisebox{2mm}%
{\parbox{.6\textwidth}{#1}}}
```

標題文字置於 `\parbox` 內，並上移 2mm。使用 `\parbox` 指令的目的是為了處理多

行之標題。因為文字置於 `\parbox` 內，不管標題文字有多少行，第 2 行開始將與上一行切齊，不至於凸出至版面左緣。

利用以上指令，若某書第 2 章之標題是以下列指令輸入：

```
\usepackage{type1cm}
\ctxfdef{\chapter}{\ctxfm}{\ctxfr}
...
\chapter{ 國際貿易與\ 國際借貸市場}
```

排版之後，章標題如下：

2 國際貿易與 國際借貸市場

請注意，鍵入標題文字時須自行拆為兩行。另外，因為數字字體放大為 45pt，故我們須在全文設定區引用 `type1cm` 巨集套件，否則 \TeX 會出現找不到字體之錯誤訊息。

使用 \TeX 指令排版章節標題時，章節會自動編入號碼。以章標題為例，標題編號是以 `\thechapter` 代表，其來源是 `chapter` 計數器。具體言之，若本章為第 7 章， \TeX 自 `chapter` 計數器中取出“7”，設定以阿拉伯數字格式編排。若章編號要改以大寫羅馬字，僅須在 `\titleformat` 指令之前加入下列指令即可：

```
\renewcommand{\thechapter}{\Roman{chapter}}
```

以上是章標題的例子。底下我們舉幾個例子說明節標題之設計，這些指令大多也可以應用於排版章或者小節標題。第一個例子是 `running-head` 標題：

```
\usepackage{titlesec}
\ctxfdef{\section}{\ctxfm}{\ctxfr}
\titleformat{\section}[runin]{\normalfont\sffamily}%
  {\S \thesection}{.5em}{---}
\titlespacing{\section}%
  {0pt}{1.5ex plus .1ex minus .2ex}{\wordsep}
```

如上所述，`\titleformat` 指令共有 7 個選項，其中第 2 與第 7 選項為非強制性，可以省略。第 1 選項為 `\section`，表示欲更改節標題。第 2 選項設定標題之排版方式，本例使用 `[runin]` 設定。在洋文書排版術語中，這稱為 `running-head`，表

示節標題文字並不單獨排成一行，而是直接置於本節第一行文字之前端。第3個選項內之指令同時控制節數字編號與標題文字。本例中，此選項內有兩道指令，`\normalfont` 先將英文與數字回復標準字體，`\sffamily` 則選用 sans serif 字體。

第4選項是排版章節編號之指令。本例中之指令為：

```
{\S\ \thesection}
```

其作用是在節數字編號之前加上裝飾符號 \S ，其後空一格，最後再加上節編號 `\thesection`。以某書之 7.5 節為例，數字編號將排為： \S 7.5。第5選項控制節編號與文字標題之間距。若數字編號與標題文字排於同一行，間距表示水平距離，本例間距為 `{0.5em}`。第6選項內是用於控制標題文字。譬如，若要在標題文字之前或上方加入任何線條或符號，指令應置於此選項內。本例中，此選項空白。最後一選項內為標題文字後之控制指令，此例之指令為 `[---]`，因此標題文字之後將加上一短橫線。排版結果如下：

\S 7.5 章節標題— 一般文稿通常區分章節，章節的編排設計必須前後一致。L^AT_EX 提供現成的指令以編排章節；一般而言，這些指令已可滿足大多數人的需求。...

除了 `\titleformat` 指令外，本例中另使用 `\titlespacing` 指令控制間距。如表 17.1 所示，此一指令有 5 個選項，第1選項也是選定所要更改的章節標題。第2選項設定「左邊空白」(left space)；「左邊空白」的意義因 `\titleformat` 第2選項之不同而異。本例中以 `[rumin]` 排版節標題，此時左邊空白指的是標題前之內縮 (indent) 空白。若 `shape` 為 `wrap` 或 `drop`，「左邊空白」設定標題文字寬度。本例子設定為 `0pt`，因此標題不內縮。

第3選項設定標題上方之間距。第4選項設定標題文字與下接正文文字之間距。若標題文字單獨為一行，此選項為垂直間距；若標題文字之後直接排版正文，此一選項控制水平間距。本例中設定為 `\wordsep`，代表英文單字間之正常間距。第5選項為附加於標題後之文字或符號，本例中從缺。

依 L^AT_EX 之原始設計，章標題之數字編號與文字標題分兩行編排；節標題則是兩者排於同一行。下一個例子是設計把節標題分為兩行，第1行排版數字編號，第2行為標題文字：

```
\titleformat{\section}[display]{\normalfont\sffamily}
  {第 \large\thesection ~ 節}{0.5em}
  {\vspace*{-15pt}\hrulefill\}[.2ex]}
```

第2選項為 [display]，指示將節編號與標題文字當作一個段落編排，因此可排版為兩行或多行。這是 \LaTeX 章標題之原始標準格式。第4選項內之 \thesection 指令代表節之編號，前面加上 \large 指令選用較大英數字體；前後又加入「第」與「節」兩個中文字。

最後的第6選項內之指令是用於排版標題文字。此一例子中，利用 \TeX 之 \hrulefill 指令畫一橫線。其前之 \vspace* 指令用以控制橫線與上面文字之間距；其後之換行指令亦加上控制指令以設定行距。再以某書7.5節為例，排版後節標題將為：

第 7.5 節

章節標題

一般文稿通常區分章節，章節的編排設計必須前後一致。 \LaTeX 提供現成的指令以編排章節；一般而言，這些指令已可滿足大多數人的需求。...

本例並未使用 \titlespacing 指令，因此所有間距將使用內定值。標題之中文字則選用粗黑體。

17.2.3 嵌入標題

第3個例子係西文排版中所謂的 drop 標題，本書稱之為「嵌入標題」。「嵌入標題」之設定通常用於節或小節，標題並非單獨成一行，而是直接置於節段落前端之左沿，首段文字的前兩三行則內縮一些。一個簡單的例子如下：

7.5	一般文稿通常區分章節，章節的編排
章節標題	設計必須前後一致。 \LaTeX 提供現成
	的指令以編排章節；一般而言，這些
	指令已可滿足大多數人的需求。...

設計標題之指令如下：

```
\titleformat{\section}[drop]%
  {\normalfont\fontseries{b}\selectfont\filright}%
  {\thesection}{.6em}{\mbox{}\\[5pt]}
\titlespacing{\section}%
  {5pc}{1.5ex plus .1ex minus .2ex}{1.5pc}
```

第2選項 [drop] 指示選用嵌入標題。第3選項最後一個指令為 `\filright`，作用和 \LaTeX 之 `\raggedright`（靠左編排）類似，但取消英文單字之音節斷字 (hyphenation) 功能。第6選項內指令為 `\mbox{}\\[5pt]`，作用是換行，並加大行距 5pt。因此，標題文字將排於編號之下。如果只有換行指令，排版時將出現錯誤訊息，因此我們先加入 `\mbox{}`，目的是填入一不占空間之虛擬文字。

使用 [drop] 選項時，`\titlespacing` 指令第2選項即設定標題文字所占空間。此例中設定之空白為 5pc。pc 是 \TeX 衡量長度的單位之一，長度大約等於 0.42 公分，請見表 7.1 (頁 97) 之說明。因為每一節之標題長度不一，若設定之長度小於標題文字之長度，標題將自動排版為兩行甚或三行。

與 drop 格式類似的是 wrap。此一格式會自動計算並調整標題寬度。若標題寬度小於給定之長度，巨集指令自動把標題空間縮小；反之，若大於給定之長度，標題也是自動拆為兩行或多行。一個簡單的例子如下：

```
\titleformat{\section}[wrap]
  {\normalfont\fontseries{b}\selectfont\filright}
  {\thesection}{.6em}{ }
\titlespacing{\section}
  {8pc}{1.5ex plus .1ex minus .2ex}{1.5pc}
```

以上指令選用 wrap 格式，`\titlespacing` 指令第2選項設定標題寬度為 8pc。輸入標題之前，我們先在全文設定區選用中文仿宋體，輸入指令時，我們自行將文字拆為兩行：

```
\ctxdef{\section}{\ctxff}
...
\section{ 章節標題\\ 排版方法}
```

排版時， \TeX 自動計算標題之寬度，並在右邊留下一點空白：

7.5 章節標題 一般文稿通常區分章節，章節的編排
 排版方法 設計必須前後一致。L^AT_EX 提供現成
 的指令以編排章節；一般而言，這些
 指令已可滿足大多數人的需求。...

17.2.4 序文與索引之標題

排版書籍時，正文每一章之標題是以相同格式編排；但目錄、序文、附錄、索引等之標題可能採用不同格式。譬如，正文每一章之標題通常編上號碼，但序文、索引之標題通常不編號，因此其標題之編排需另作處理。

舉例言之，若使用 `\tableofcontents` 指令排版目錄，依 L^AT_EX 之設定，目錄之標題是以 `\chapter*` 指令編排，* 號表示不編號。如果我們使用 `\titleformat` 指令設計章標題之編排，則目錄頁之標題可能須另行設計。文稿內，`\titleformat` 指令可多次使用，因此，我們可以在目錄之前另行設計其標題格式。同樣的，序文之標題通常不加上編號。L^AT_EX 並無排版序文之專用指令，但我們也可以使用 `\chapter*` 指令編排序文標題。

不過，`\chapter*` 指令有一個問題，其所排版的標題不會自動納入目錄中。欲解決此問題，我們可使用下列指令：

```
\addcontentsline{toc}{ sec-name}{text}
```

以上指令中，*text* 為欲出現在目錄中之文字。*sec-name* 為章節指令名稱，但前端不加上反斜線，例如 `chapter`，`section`。如果我們以 `\chapter*` 指令排版序文標題，則指令之前加上：

```
\addcontentsline{toc}{chapter}{\ctxfbb 序文}
```

排版後，「序文」兩字將出現於目錄內，並以粗黑體編排。

17.3 titletoc 巨集套件

L^AT_EX 排版目錄的指令是 `\tableofcontents`，其格式是固定的。本書之目錄即使用現成格式排版。欲自行設計目錄格式，可使用 `titletoc` 巨集套件，這是 `titlesec` 作者 Javier Bezos 的另一項創作。使用 `titletoc` 巨集套件時，我們仍然是

<div style="display: flex; justify-content: space-between; align-items: center;"> <i>left</i> ← </div> <p style="margin-top: 10px;">5 消費函數</p> <p style="margin-top: 5px;">5.1 跨期的消費選擇</p> <p style="margin-top: 10px; font-size: small;">借貸市場, 132 ▶ 儲蓄, 134, ▶ 預算限制式, 136</p>	<div style="display: flex; justify-content: space-between; align-items: center;"> <i>right</i> ← </div> <p style="margin-top: 10px; font-size: small;">130</p> <p style="margin-top: 5px; font-size: small;">131</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

圖 17.2: 目錄圖示

以 `\tableofcontents` 指令排版目錄, 但另以 `titletoc` 巨集套件之指令設計目錄之排版格式。

介紹指令之前, 我們先了解目錄之一般樣式。圖 17.2 是目錄排版的一個例子, 其中包含第 5 章, 5.1 節, 及其下的三個小節。章與節目錄之格式與 \LaTeX 內定格式相同, 但小節目錄是新格式。本例中, 左右兩直線代表文字版面邊沿。右邊小版面 (*right*) 排版章節出現處之頁碼, 其空間大小可由下列指令設定:

```
\contentsmargin{8mm}
```

以上指令設定右邊小版面之空間為 0.8 公分。

左邊小版面大小之設定並無單獨的指令, 而是由 `\titlecontents` 指令所控制, 指令之格式請見表 17.2。舉例言之, 若使用 \LaTeX 之 `book` 文件類別排版, 其目錄中之節標題大約是以下列指令定義:

```
\titlecontents{section}
  [1.5em]{}
  {\contentslabel{2.3em}}
  {\hspace*{-2.3em}}
  {\titlerule*[1pc]{.}\contentspage}
```

第 1 行指令指示要更改 `\section` 之目錄排版方式。第 2 行方括號內為 `1.5em`, 此設定左邊小版面 (*left*) 空間之大小。其後之大括號內空白, 故整個目錄項之排版採內定值, 並無額外之排版控制。第 3 行指令設定排版節目錄文字之前應空出 `2.3em` 空間, 並填入節編號; 之後, 則自動加入節標題文字。第 4 行指令適用於無編號之節標題, 因為沒有編號, 故往左移 `2.3em`, 以與上一個節標題對齊。第 5 個指令設定細點之間距 `1pc`, 行末則填入頁碼。

圖 17.2 例子中之小節目錄較特別, 小節不加上編號, 目錄文字其後不加上細點直線, 而是緊接著排出頁碼。兩個小節目錄文字之間以一右三角形區隔。若直

表 17.2: `titletoc` 巨集套件指令

`\titletoc{ section}[left]{above}{before with label}{before without label}`
`{filler and page}[after]`

- *section*: 欲重新設計之指令, 如 `chapter`, `section`, `subsection` 等, 請注意不須加入反斜線。
- *left*: 左邊小版面之空間大小, 參見圖 17.2。
- *above*: 控制整個目錄項之排版指令。譬如, 本目錄項與上一個目錄項之間距可於此設定。
- *before with label*: 排版本目錄項文字部分之前所加入之控制指令。我們所設計排版之目錄項可能含章節編號 (label), 也可能不含。以節標題為例, `\section` 指令排版含編號之標題, `\section*` 指令之標題則不含編號。此項指令是用於含編號之情況。如同 `\titleformat` 指令 (表 17.1, 頁 328) 之 *before* 選項, 此選項可能含一個或多個指令, 最後一個指令可帶有一個代表標題文字之參數。
- *before without label*: 同上一項, 但適用於不含編號之標題。
- *filler and page*: 排版連接之細點及頁碼。
- *after*: 排版於目錄文字後之符號。此非強制選項, 可不加入。

控制指令

- `\contentsmargin{ right}`: 設定版面右沿空白之大小。
 - `\contentslabel{ length}`: 在本行前端空出 *length* 大小, 並排版章節標題編號。
 - `\contentspage` : 在本行末端排版頁碼。
 - `\thecontentlabel` : 章節標題之編號。
 - `\titlerule*[width]{text}`: 將本行填入細點, 細點以 *text* 選定, 點距以 *width* 設定。若不設定點距, 內設值為其本來之間距。
-

接排出全黑色右三角形, 符號顏色太深, 故我們設定讓右三角形以灰階色排出。這是 `cwTeX` 所提供的書籍排版例子所使用之設計, 例子之說明請見 18.2 節 (頁 349)。底下列出 `\titlecontents` 指令之設定, 並簡單說明其意義。

因為使用三角形的符號, 故 1–3 行指令首先定義排版右三角形之指令。第 1 行指令定義 `half` 顏色為 50% 之灰階, 使用於 2–3 行的指令中。因為小節不排版編號, 故我們使用加上 * 號之 `\titlecontents*` 指令。第 1 選項為 `subsection`, 指示重新設計小節目錄之排版。第 2 選項 `[14mm]` 設定左邊小版面之空間, 第 3

選項選用字體及行距，並加大行距 0.6mm。第 4–5 選項空白，使用內定值。本例中，小節目錄不排版編號，故第 4 選項不適用。第 5 選項為空白，故直接排版標題文字。標題文字後之排版由第 6 選項 *filler and page* 控制，先加上逗點，再填入頁碼。最後，右三角形符號則設計於第 7 個選項內，符號前後各留出一點空白。

```
\definecolor{half}{gray}{.50}
\newcommand{\triang}{\fboxsep=0pt
\textcolor{half}{\blacktriangleright}}
\titlecontents*{subsection}
[14mm]{\fontsize{10.95}{16pt}\selectfont\vspace*{0.6mm}}
{}{}
{, \thecontentspage}[ ~\triang{~ }
```

`titletoc` 巨集套件還可以作其他排版設計，例如，我們可在每一章前端排版該章之目錄，或者在目錄內加入裝飾圖形。Bezos 所寫的兩個巨集套件功能強大，但是指令說明不太清楚。不過，說明檔案內提供不少例子；仔細模索，不難由之設計你所要的格式。

17.4 fancyhdr 巨集套件

排版一般文稿，我們通常在版面上方或下方加入頁碼。若是排版書籍，除了頁碼之外，我們通常還會在頁眉與頁足排版章節標題。在西文排版中，頁眉與頁足之排版設計是所謂的頁面格式 (page style)。L^AT_EX 提供 `\pagestyle` 指令以頁排版頁面格式。若我們僅是要排版出頁碼，使用 L^AT_EX 之指令已足夠。但是，若頁眉與頁足要排版中文章節標題，L^AT_EX 之指令可能出現問題，無法正確排版，請參見 7.6 節 (頁 120) 之說明。

欲正確排版中文稿之頁面格式，我們可使用 Piet van Oostrum 所寫的 `fancyhdr` 巨集套件。在介紹指令之前，我們先說明基本概念。圖 17.3 說明版面上頁眉與頁足之位置。在洋文書的排版中，以 `book` 文件類別所排版的書籍，左右頁之版面格式不同。通常，左邊頁碼為偶數 (even)，右邊為奇數 (odd)。圖中之 LH-even 代表偶數頁左邊頁眉，其中 LH 代表左邊頁眉，even 代表偶數頁。同理，CF-odd 代表奇數頁中央頁足。若以短文類別排版文稿，但不選用 `[twoside]`，則每一頁皆視為右頁。下一小節先說明左右頁版面相同之情況。

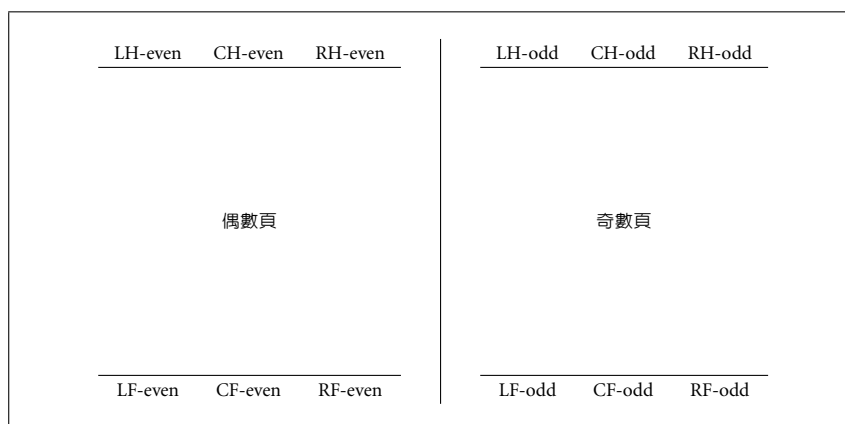


圖 17.3: 頁眉與頁足版面圖示

17.4.1 單雙頁相同版面

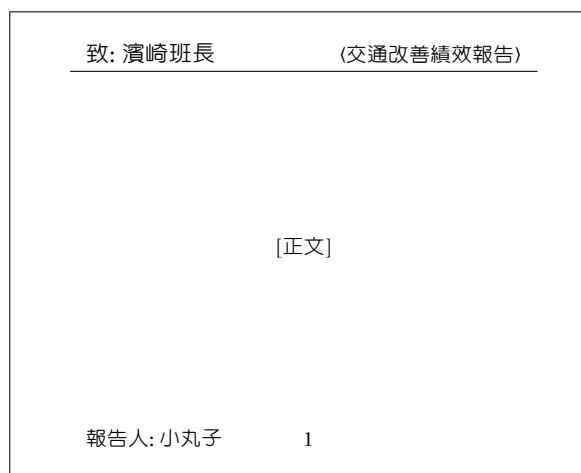
上面說明, \LaTeX 之 `\pagestyle` 指令提供 `plain` 等 4 種格式, `fancyhdr` 巨集套件則提供另一種頁眉/頁足格式, 稱為 `fancy`, 其內容接近 `headings`, 但很容易重新設定。欲使用 `fancyhdr` 巨集套件, 首先在全文設定區鍵入下列 2 行指令:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

第 2 行指令指示以 `fancy` 格式排版頁眉/頁足, 因此, 接下來我們須輸入定義 `fancy` 格式之指令。

圖 17.4 是短文 `article` 文件類別的例子, 其左右頁格式完全相同。版面上頁眉與頁足共有 6 個地方可供排版。假設小丸子要向濱崎班長報告交通改善之績效, 報告之左上角排版報告對象, 右上角排版題目, 左下角排版作者, 頁碼則排版於頁面下方中央。定義格式之指令可置於文稿任何地方, 不過, 若整篇文稿之格式都相同, 定義指令可置於全文設定區。

本例中, `\lhead` 指令用於排版頁眉左欄, `\rhead` 用於排版右欄, `\lfoot` 則用於排版頁足左欄, 餘此類推。`\chead{}` 指令表示頁眉中央為空白; `\cfoot` 指令內之 `\thepage` 指令用於排版頁碼。倒數第 2 行指令為 `\headrulewidth`, 其目的是在頁眉畫一粗細 0.4pt 之橫線, 長度為 `\headwidth`。最後一行之 `\footrulewidth`



```

\lhead{\ctxfr 致: 濱崎班長}
\chead{}
\rhead{\ctxfr (交通改善績效報告)}
\lfoot{\ctxfr 報告人: 小丸子}
\cfoot{\thepage}
\rfoot{}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0pt}

```

圖 17.4: fancyhdr 巨集套件應用例子

定為 0pt, 因此頁足沒有長線。本例中, `\rfoot` 為空白; 如果要在頁足右欄排版當天日期, `\rfoot` 指令內容應改為:

```
\rfoot{\today}
```

如果填入 `\jobname`, 則排出檔案名稱。

以上所定義之 `fancy` 格式, 適用於文稿中大部分版面之頁眉與頁足。但某些特定的頁面, 頁眉與頁足須使用特別的格式。譬如, 以 `book` 文件類別排版之文稿可使用 `\chapter` 指令排版章標題, 而根據洋文書之排版規範, 章標題頁通常不排版頁眉與頁足。另外, 一般文稿之題目頁之頁面設計也與普通頁面不同。如果某頁只要排版頁碼, 其餘的資訊都不出現, 僅須在 `\begin{document}` 指令之後加入下一行指令即可:

```
\thispagestyle{plain}
```

不過,若文稿之第1頁已使用 `\maketitle` 指令排版題目,以上之指令事實上可以省略,因為 \TeX 會自動在 `\maketitle` 指令之後加入上述指令。

如果標題頁連頁碼都不要,應使用 `empty` 格式:

```
\thispagestyle{empty}
```

一般正文的頁面是以 `fancy` 格式設計。若標題頁版面之頁眉/頁足要與正文其他各頁相同,則標題頁可以下列指令選用相同格式:

```
\thispagestyle{fancy}
```

請注意,若文稿使用 `\textwidth` 指令設定正文寬度,該項指令應置於 `fancyhdr` 巨集套件設定指令之前,否則頁眉之位置與正文文字方塊可能無法對齊。

17.4.2 單雙頁版面不同

若單頁與雙頁之版面格式不同,則單雙頁版面都須定義。最基本的兩項設定指令如下:

```
\fancyhead[RO]{...}
\fancyfoot[CE]{...}
```

第一行指令用於設定頁眉之格式,第二行指令則用於設定頁足之格式。方括號內須填入所欲設定之欄位。本例中, `RO` 代表奇數頁 (odd page) 右欄, `CE` 代表偶數頁 (even page) 中間欄。同理 `LE` 為偶數頁左欄, `LO` 則代表偶數頁左欄。

圖 17.5 的例子比圖 17.4 更複雜一些。開始設定之前,我們先清除各欄位原有之內容,這是第 1–2 行指令的作用;第 3 行開始再填入各設定文字。假設偶數頁頁眉左欄 (`LE`) 與奇數頁頁眉右欄 (`RO`) 要排為「致:濱崎班長」,下列指令第 3 行即作此項定義。同理,4–5 行指令之作用分別為:偶數頁頁眉右欄與奇數頁頁眉左欄為 (交通改善績效報告);偶數頁頁足左欄與奇數頁頁足右欄為「報告人:小丸子」。單雙數頁之頁足之中間欄皆為頁碼;最後兩行指令則在頁眉下沿及頁足上沿各畫出 0.4pt 粗細之直線,長度為 `\headwidth`。

上面曾說明,我們可使用 `\thispagestyle{plain}` 指令讓頁眉空白,頁碼排於版面下方中央。使用 `book` 文件類別時,每一章開頭之 `\chapter` 指令出現處,

```

\pagestyle{fancy}
\fancyhead{}
\fancyfoot{}
\fancyhead[RO,LE]{\ctxfr 致：濱崎班長}
\fancyhead[RE,LO]{\ctxfr (交通改善績效報告)}
\fancyfoot[RO,LE]{\ctxfr 報告人：小丸子}
\fancyfoot[CO,CE]{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}

```

圖 17.5: fancyhdr 巨集套件應用例子之二

TeX 自動加入上述指令，因此我們不須再自行輸入。雖然 TeX 會自動將每一章之首頁以 `plain` 格式編排，但其格式可能不是我們所想要的。譬如，學術期刊內每一篇論文之首頁的正下方通常會排出版權聲明。如果我們是以 `book` 文件類別排版整本期刊，則每一章首頁之格式必須調整。fancyhdr 巨集套件提供 `\fancypagestyle` 指令，其功能即是重新設計 `plain` 之格式。

若要把首頁之頁足中央改為版權聲明，我們須重新定義 `plain` 之格式。若使用 fancyhdr 巨集套件，僅須在全文設定區輸入下列指令即可：

```

\fancypagestyle{plain}{%
\fancyhf{} % clear all header and footer fields
\fancyfoot[C]{\copyright} \textit{TER}}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}}

```

第 1 行指令表示要重新定義 `plain` 格式，第 2 行指令清除頁眉與頁足原有之內容，使之變為空白；第 3 行指令設定頁足中央排版出下列標誌：

© TER

第 4-5 行設定頁眉/頁足都不要橫線。

17.4.3 頁眉與頁足之章節標題

書籍排版常把章節標題置於頁眉或頁足，一般洋文書的習慣是把章標題排於雙數頁之頁眉或頁足，節標題排於單數頁。article 文件類別不能使用 `\chapter`

指令, 但若引用 `twoside` 選項, 則節標題通常排版於雙數頁之頁眉或頁足, 小節標題則排於單數頁。

\LaTeX 會自動記錄章節標題, 因此要將章節標題排於頁眉並不困難。最直接的方法是使用 `\leftmark` 與 `\rightmark`。在 `book` 文件類別下, 前一指令記錄章標題; 後一指令記錄單數頁之資訊。相對的, 在 `article` 文件類別下, `\leftmark` 存錄節標題, `\rightmark` 則記錄小節標題。當頁碼增加, 由第 1 節變為第 2 節, 或由第 3 章變成第 4 章時, 以上兩項指令所記錄之內容也隨著更新。

以 `book` 文件類別排版書籍時, 若偶數頁頁眉之左欄要排版章標題, 奇數頁右欄要排版節標題, 而頁碼要排在偶數頁頁眉之右欄與奇數頁頁眉之左欄, 則 `fancy` 格式可定義如下:

```
\fancyhead{}
\fancyfoot{}
\fancyhead[LE]{\leftmark}
\fancyhead[RO]{\rightmark}
\fancyhead[RE,LO]{\thepage}
```

加入以上指令之後, 版面上方會出現一條橫線。這是 `fancyhdr` 巨集套件之原始設定。

在 `book` 文件類別下引用 `fancyhdr` 巨集套件時, 若僅選取 `fancy` 格式而不加入任何設定指令, 內定之格式如下:

```
\fancyhead[LE,RO]{\slshape \rightmark}
\fancyhead[LO,RE]{\slshape \leftmark}
\fancyfoot[C]{\thepage}
\headrulewidth=0.4pt
\footrulewidth=0pt
```

第 1 行指令設定偶數頁頁眉左欄與奇數頁右欄為節編號與標題; 第 2 行指令設定偶數頁右欄與奇數頁左欄為章編號與標題。以上的內定格式雖然完美, 但中文稿並不適用。以第 7 章為例, 頁眉之標題除了章名之外, 還會自動加上英文 `CHAPTER` 與數字編號, 格式為 “`CHAPTER 7.`”。不過, 以上的格式也可以重新設定。

\LaTeX 是以下列三個指令儲存頁足/頁眉之章節編號與標題文字格式:

```
\chaptermark
```

`\sectionmark`

`\subsectionmark`

以上指令分別儲存章、節、小節之頁眉/頁足標題。欲重新設計頁眉/頁足之標題格式，我們須重新定義以上的指令。第 1 個指令 `\chaptermark` 用於設定章標題之格式；後兩個指令分別用於節與小節標題。例如，若要去掉章編號之前的英文字“CHAPTER”，須在全文設定區加入指令：

```
\renewcommand{\chaptermark}[1]{\markboth{\thechapter.\ #1}{}}
```

以上指令使用 `\markboth` 指令重新設定 `\chaptermark` 指令所定義頁眉/頁足內之章標題格式。`\markboth` 指令存放兩項資訊，分別置於指令之後兩對大括號內。在 `book` 文件類別中，第一對大括號內存放欲出現在左頁章標題處之文字；第二對大括號內則存放欲出現在右頁節標題處之文字。

本例中，節標題不更動；因此第二對大括號內為空白。章標題處之文字可能包括章編號與標題，另外，我們也可能加入諸如“Chapter”之文字。ET_EX 將章編號儲存於 `\thechapter` 指令內；而 `\chaptername` 指令可用於定義英文字“Chapter”。若某章為第 7 章，因此該章範圍內，`\thechapter` 之值即為“7”。在中文稿件中，我們不想加入“Chapter”文字，因此 `\chaptername` 可省略不用。本例中，`\thechapter` 指令之後緊接一英文句點，其後以 `_` 指令加入一空格，最後的 #1 參數則代表章名。

經過以上之重新設計，`\chaptermark` 指令即將左頁之章標題定義為：

7. 文稿結構與章節設計

如果頁眉之章編號要改為「第 7 章」，上一指令應更改如下：

```
\renewcommand{\chaptermark}[1]{\markboth{第\thechapter 章\ #1}{}}
```

其中，中文字之字體與大小尚可以中文字體指令設定。

出現在頁眉與頁足中之英文/數字，`fancyhdr` 巨集套件會自動選用特定字體，但是這些指令並不能改變中文字體或其大小。欲改變中文章標題之字體，可於下 `\chapter` 指令時直接鍵入字體指令，或者以 `\ctxfdef` 指令作全文設定。若章標

題是以中文粗黑體 20 點編排，則章標題出現於頁眉時其大小即為 20 點字體。欲解決此一問題，`\chapter` 指令應加入選項，例如：

```
\chapter[ 文稿結構與章節設計]{文稿結構與章節設計}
```

同時，全文設定區再以 `\ctxfdef` 指令選用中文字體。經以上設定， \TeX 將取用方括號中之內容排版頁眉/頁足。

事實上，選項方括號內之文字可以和大括號內之標題文字不同。譬如，若章標題文字太長，利用選項可使出現於頁眉內之文字簡短一些。另外，若使用加 * 號之章節指令，如 `\section*` 等，標題內容不會自動編入頁眉中，但我們可使用選項指令編排頁眉/頁足。

除了章標題之外，單數頁之頁眉/頁足上之節與小節標題也可以使用類似的指令更改。以節標題為例，儲存節標題之指令為 `\sectionmark`；編號則儲存於 `\thesection` 指令內。舉例言之，若頁眉之節標題只要文字不要編號，指令為：

```
\renewcommand{\sectionmark}[1]{\markright{#1}}
```

請注意，改變節標題須使用 `\markright` 指令，而該指令僅有一選項。同理，以類似的指令重新定義 `\subsectionmark` 之內容，即可變更頁眉/頁足之小節標題之格式。

本書之頁眉為空白，頁碼與章節標題全部置於頁足。使用之指令列舉於下，以供參考：

```
\fancyhf{}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{%
  {\markright{\thesection~ #1}}
}
\fancyfoot[LE]{\cdots\the page\cdots\rule{4mm}{0pt}\leftmark}
\fancyfoot[RO]{\rightmark\rule{4mm}{0pt}\cdots\the page\cdots}
\renewcommand{\headrulewidth}{0pt}
```

以上指令在左頁頁足之左欄排版章標題，但不編號；右頁頁足之右欄則排版節標題。其中，`\rule{4mm}{0pt}` 是用以在頁碼與標題之間產生 4mm 之間距；頁碼兩旁之小點是以 `\cdot` 指令排版。

以上所介紹的指令適用於排版正文各章節之頁眉/頁足，而文稿正文以外的

頁眉/頁足之排版可能須另作處理。以本書為例, 1–22 章依序編號, 章標題與節標題分別出現於左右頁下端。但是,「序」是以 `\chapter*` 指令編排; 第 22 章之後的參考書目與索引也是以 `\chapter*` 指令編排。序文標題並編號, 內容也不分節, 索引亦然, 因此正文各章之頁眉/頁足格式並不適用。 `fancyhdr` 之設定指令可在文稿內多次使用, 故排版時, 我們設計兩種標題格式, 一種用於排版正文各章之標題, 另一種則用於排版序、參考書目、與索引。

巨集套件 `fancyhdr` 所提供的功能甚多, 以上僅介紹其中較常用者, 完整的功能介紹請見該巨集指令之說明檔。該檔案說明如何讓頁眉長度大於正文方格之寬度; 也說明如何讓空白頁之頁眉/頁足空白。有關於 `\leftmark` 與 `\rightmark` 指令之詳細說明, 請見 [Goossens et al. \(1994\)](#), 4.3.1 節。最後, 圖 7.4 (頁 111) 的例子使用了 `fancyhdr` 巨集套件, 例子檔案置於 `c:\texmf\cwtex\examples` 檔案夾內。請嘗試編排, 以了解各項指令之作用。

18 書籍排版

排版出一本高品質的書籍是 \LaTeX 系統的主要目標, 對排版者而言, 則是最大的挑戰。除了封面之外, 一本書從封面內頁開始至少包括: 封面內頁 (half-title), 版權頁, 目錄, 序言, 正文各章節, 參考書目, 索引等。性質特殊的書籍還有有特別的需求, 例如教科書裡, 每一章可能須排版該章之簡目, 每章結束須排版摘要與習題, 書末則須提供習題解答。

本章將以一個例子說明書籍排版的過程。此一例子含前言及內文兩章, 全部檔案置於 `c:\texmf\cwtex\examples\book` 檔案夾內。我們還將介紹編排索引的方法, 以及如何在版面四角印上裁切記號 (crop marks), 以方便書籍的裝訂作業。在介紹例子內之指令之前, 我們須強調, 高品質排版的先決條件是設計與美感, 排版系統只是工具而已。排版系統好比照相機, 有人手持 Nikon F5 相機, 照出來的相片卻比傻瓜相機還不如。如果真的要排出一本好書, 請先了解書本排版的基本觀念。欲了解排版, 我們推薦 [Bringhurst \(1996\)](#)。如果不想花錢買書, [Wilson \(2001\)](#) 也值得一讀。

18.1 排版設計

排版書籍或長篇文稿時, 最重要的是全書體例一致。欲達到此點, 我們應儘可能使用巨集指令或巨集套件。譬如, 每章標題、圖表標題、頁眉與頁足等, 若不使用巨集指令處理, 很容易出現前後不一致的情況。目錄或索引的排版也是如此, 由 \LaTeX 所提供的指令計算各章的頁碼, 可保證不出錯。

另外一個考慮重點是, 排版指令應儘可能保持修改的彈性。例如, 若某書籍共有15章, 為方便起見, 我們應採分章排版的方式進行。第1章取名為 `chap1.ctx`, 第2章取名為 `chap2.ctx`, 等等。但是, 更好的是依每章之性質取檔名, 譬如第1章取名為 `intro.ctx`, 第2章取名為 `demand.ctx`。若採用前一方式, 萬一未來在第2章之前要加入新的一章, 取名將產生困擾; 後一方式則無問題。

同理，書本內若須交叉引述，最好是使用 `\label` 與 `\ref` 指令。例如，第 8 章的標題之後可加上 `\label{demand}` 設定指令，而未來某章須徵引此章之章節時，僅須使用“參見第 \ref{demand} 章”指令即可。若直接排版成“參見第 8 章”，則萬一第 8 章之前又加入新的一章，全書徵引章節之行文都須修改。編排書末的索引也一樣，我們應使用 \TeX 所提供的索引編排指令，切勿自行找出頁碼，再手動編排。

排版工作到了最後階段，有幾項情況不可避免地須手動調整。第一種情況是，某些行文字右端會超出版面右沿。 \TeX 會儘可能把維持版面右沿切齊，但有些情況實在是不可能處理的完美的。此時，我們必須在上下句子中添加或刪減文字，或者改寫句子，才能解決問題。

第二種情況是，某些頁的版面行距可能顯得寬鬆，或特別緊貼。產生狀況的原因有多種可能性。可能性之一如下： \TeX 會避免讓節標題單獨出現在一頁底端。因此，若排完本節最後的文字之後，本頁所餘空間不足以排版下一節標題，則 \TeX 將把下一節標題移到下一頁，本頁行距即顯得特別大。此時，我們可以在本頁最後段落末端加上 `\newpage` 或者 `\clearpage` 指令。以上指令使本頁以正常行距排出，但末端將有一些空白出現。

必要時，我們也可縮減或加大本頁文字版面之高度。例如，欲將本頁排版之行數減少一行，可在本頁文字段落之間加入底下指令：

```
\enlargethispage{-\baselineskip}
```

我們也可以用 `\enlargethispage{0.5cm}` 指令，將文字版面高度 (`\texttheight`) 增加 0.5 公分。

編排長篇書籍時，必然會使用較多的字體。中文因為字數眾多，編排長篇書籍時，有時候會因為使用較多字型，而無法編排；或者無法轉換成 PostScript 檔案。以排版本書之實際經驗來說，我們須作下列調整：

- `c:\localtexmf\miktex\config\miktex.ini` 檔案內儲存字型記憶體容量之參數 `font_mem_size=200000` 須加大為 400000。
- `dvips` 無法轉換出 `.ps` 檔案，我們必須改用 Omega 系統之 `odvips.exe`。另外一個辦是改用 MiKTeX 2.2 版之 `dvips.exe`。此版本目前仍為測試版，請見 [4.3.1 \(頁 53\)](#) 之安裝說明。

18.2 書籍排版例子

排版書籍通常使用 `book` 文件類別。7.2.5 節 (頁 110) 曾介紹一個以 `book` 文件類別排版研究生論文的例子。本節將介紹排版書籍的例子, 此例子排版一本經濟學教科書, 全書包含封面內頁、版權頁、目錄、序、正文 (兩章)、習題、參考文獻、習題解答、與索引。全部的檔案置於 `c:\texmf\cwtex\examples\book` 檔案夾內。

排版書籍, 我們首先須考慮整本書的文稿結構。整本書可分 (正文之前) 目錄與序文、正文內容、以及 (正文之後的) 參考文獻與索引等。在 \LaTeX 中, 這三部分稱為 `\frontmatter`, `\mainmatter`, 及 `\backmatter`。我們可以在文稿中加入以上三道指令以區分三個部分。使用這些指令的好處之一是它們可以幫忙設定頁碼之字體。譬如, 如果在正文的排版指令與文字之前加入 `\mainmatter` 指令, 則目錄與序文之頁碼將以小寫羅馬字排版, 正文部分則用阿拉伯數字。不過, 本例子並未使用這三道指令, 因此頁碼是從頭開始以阿拉伯數字編排。

除了文稿結構之外, 我們須作版面設計。此處所謂的版面設計, 包括版面尺寸、章節標題、以及頁眉與頁足之文字。 \LaTeX 本身有標準的版面設計, 但我們也可以利用第 17 章所介紹之之巨集套件變更設計。自行設計版面時, 首先須注意正文本身與正文前後之目錄與索引之版面可能不同。通常, 正文各章各有編號, 如第 1 章、第 2 章等, 但目錄、參考文獻並無編號, 故標題可能採不同的設計。

本例子採用分章編排的方法, 書籍之內容所含的兩章分別取名為 `w-gdp.ctx` 與 `w-growth.ctx`, 序文為 `w-intro.ctx`。序文之前的封面內頁及版權頁之排版則置於 `w-htitle.ctx` 檔案內。以上 4 個檔案之外, 另有一個主檔案 `ecobook.ctx`, 其內容全文列於圖 18.1 以供參考。4 個子檔案是以 `\include` 與 `\includeonly` 指令納入主檔案內, 我們先說明其用途。一開始排版時, 各章之內容可分開排版, 到了接近完成階段, 才創造 `ecobook.ctx` 主檔案, 以 `\include` 與 `\includeonly` 指令將各子檔案引入。

在主檔案內, `\include` 指令應把全書各章之檔案全部引入, 但排版時, 利用 `\includeonly` 指令我們可以僅排版目前欲修正之檔案。請特別注意, 各子檔案須以 `.ctx` 為延伸檔名; 但是, 延伸檔名卻不可鍵入於指令內。本例中, 全部的子檔案都納入在主檔案內。現假設文稿仍在修正階段, 而 `\includeonly` 指令僅引入 `w-intro.ctx` 與 `w-gdp.ctx` 兩檔案。並假設在前一輪排版全文時, 已產生了全文的頁碼、圖表目錄、數式編號等, 則重新以 `latex` 排版時, 程式僅處理

```

1 \input{w-macro}
2 \renewcommand{\figurename}{\ctxfr 圖}
3 \renewcommand{\tablename}{\ctxfr 表}
4 \renewcommand{\contentsname}{\ctxfr 目錄}
5 \renewcommand{\indexname}{\ctxfr 索引}
6 \renewcommand{\bibname}{\ctxfr 參考文獻}
7
8 \hoffset=1cm \voffset=6cm % for 2-up
9 % \hoffset=1.5cm \voffset=3cm % for b5 paper
10
11 \ctxfdef{\chapter}{\ctxff}{\ctxff}
12 \ctxfdef{\section}{\ctxfm}{\ctxfbb}
13 \ctxfdef{\subsection}{\ctxfm}{\ctxfbb}
14 \ctxfdef{\subsubsection}{\ctxfm}{\ctxfk}
15 \ctxfdef{\paragraph}{\ctxfm}{\ctxfk}
16 \ctxfdef{\caption}{\ctxfr}
17
18 % copyright information, can be deleted
19 \fancypagestyle{plain}{%
20 \fancyhf{}
21 \fancyfoot[C]{\copyright 吳聰敏
22 \small (\today)}}
23 \renewcommand{\footrulewidth}{0pt}
24
25 \includeonly{w-htitle,w-intro,w-gdp,w-growth}
26
27 \begin{document}
28 \onecolumn
29
30 \setcounter{page}{0} %% for 2-up printing
31 \mbox{} \newpage %%
32
33 \include{w-htitle}
34
35 \nonumtitle
36 \nonumhead{{\ctxff 目錄}}
37 \tableofcontents
38
39 \include{w-intro}
40 \include{w-gdp}
41 \include{w-growth}
42
43 % Pagestyle for bibliography
44 \addcontentsline{toc}{chapter}{\ctxff 參考文獻}
45 \nonumtitle
46 \nonumhead{{\ctxff 參考文獻}}
47
48 \bibliographystyle{cw}
49 \bibliography{test}
50
51 \clearpage
52
53 % Solutions are in two-column format
54
55 \geometry{paperheight=23cm,paperwidth=16.5cm,%
56 textwidth=13cm,textheight=18.5cm,left=2cm,%
57 twosideshift=-.25cm,top=.3cm,footskip=1.75cm,%
58 marginparwidth=2cm,twoside}
59
60 \nonumtitle
61 \nonumhead{{\ctxff 習題解答}}
62 \addcontentsline{toc}{chapter}{\ctxff 習題解答}
63 \twocolumn
64
65 % Change Figure 2.3 (in chapter 2) to Figure 3.
66
67 \renewcommand{\thetable}{\arabic{table}}
68 \renewcommand{\thefigure}{\arabic{figure}}
69
70 \chapter*{ 習題解答}
71 \section*{ 第1章} \input{ans1}
72 \section*{ 第2章} \input{ans2}
73
74 \newpage
75 \addcontentsline{toc}{chapter}{\ctxff 索引}
76 \nonumhead{{\ctxff 索引}}
77 \fontsize{10.95}{16pt}\selectfont
78
79 \printindex
80 \end{document}

```

圖 18.1: 排版書籍

w-intro.ctx 與 w-gdp.ctx 兩個檔案，但上一輪排版所計算出來的頁碼與圖表目次資訊，將會納入排版文稿內。如果此次修改之後，w-gdp.ctx 這一章之頁碼增加，則原先所計算之下一章頁碼資訊將不正確。欲得正確結果，須把輔助檔案全部刪除，重新排版全書。

目錄可以利用 `\tableofcontents` 指令排版， \LaTeX 自動以 `\chapter*` 指令排版其標題。參考文獻與索引也是以同樣方法排版。相對的，序與習題解答並無現成的指令可以排版其標題，但我們也可以使用 `\chapter*` 指令自行編排。就整本書而言，我們有兩套版面設計指令，一套用於正文各章，另一套用於目錄、序、索引等。根據以上的概念，我們進一步說明圖 18.1 所示之主檔案的內容。

- 第 1 行：引入 `w-macro.ctx`，內含排版本例子所需之巨集套件及指令。此一檔案所定義的巨集指令中，有兩個內含中文字，故檔案以 `.ctx` 為附檔名，請參見 14.1.2 (頁 281) 之說明。排版時，`cwTeX` 會將檔案內之中文字轉為 \TeX 指令，檔名則改為 `w-macro.tex`。
- 2–6 行：重新定義圖、表、索引等之標題成為中文字。
- 8, 30–31 行：排版時，我們經常須印出結果校對。為了節省紙張，我們可以採用 2-up 的印法，亦即將兩頁略為縮小印於一張紙上。參見 13.5.3 (頁 274) 之說明。若採此一印法，版面位置須略作調整，這是第 8 行指令的作用。此外，因為偶數頁在左，奇數頁在右之規範，全書應從第 0 頁開始，如此一來奇數頁才會出現在紙面右邊，這是第 30–31 行指令的作用。排版完成時，請記行第 8 行指令應刪除，改用第 9 行；而 30–31 行指令則須註銷。
- 11–16 行：選擇標題內中文之字體。
- 19–23 行：在每章之標題頁正下方排出版權聲明，在初稿流通階段這可能有需要。正式出版時，這 5 行指令應註銷。
- 28 與 63 行：本書正文各章採單欄 `\onecolumn` 排版，書末之習題解答與索引則採兩欄式 `\twocolumn`。
- 33–37 行：第 33 行引入 `w-htitle.ctx` 以排版封面內頁，版權頁，及序文；第 37 行指令 `\tableofcontents` 的功能是排版目錄。上面已經說明，本例子正文各章之版面設計與目錄、索引等之版面不同。為了方便起見，我們在 `w-macro.ctx` 檔案裡定義 `\nonumtitle` 巨集用以排版正文以外各主要部分

之標題; `\nonumhead` 巨集指令則用於排版頁眉與頁足。因此, 35–36 行引用這兩道指令以排版目錄之標題、頁眉、與頁足。

- 39 行: 引入 `w-intro.ctx` 以排版序文。排版目錄時所下之 `\nonumtitle` 標題指令 (第 35 行), 仍持續適用於排版序文標題; 但頁眉與頁足之文字須改變, 故我們在 `w-intro.ctx` 檔案前端加入 `\nonumhead{\ctxff 序}` 設定指令。
- 40–41 行: 引入正文第 1 章與第 2 章之檔案。每章開頭, 我們都加入下列兩行指令:

```
\chkeven
\numtitle
```

第 1 行指令確認各章都是由偶數頁開始編排; 第 2 行指令設計正文之章標題以及頁眉與頁足之排版。為免主檔案太過複雜, 這兩道巨集指令之定義都置於另外一個存放巨集指令定義之檔案 `w-macro.ctx` 內。

- 45–46 行: 參考文獻之版面設計 (包括標題、頁眉與頁足)。
- 48–49 行: 以 `cwbibtex` 程式排版參考文獻。
- 55–63 行: 習題解答與索引都是以兩欄格式編排, 且版面之行寬加大, 故 55–58 行先以 `geometry` 巨集套件重新定義版面尺寸。原版面尺寸之設定是定義於 `w-macro.ctx` 內。60–61 行設定習題解答之版面; 67–68 行設定習題解答內之圖表編號格式。71–72 行使用 `answers` 巨集套件排版習題解答, 參見 15.6 節 (頁 307) 之說明。
- 76 行: 設定索引版面之頁眉與頁足。
- 79 行: 排版索引。L^AT_EX 排版索引的方法請見 18.3 節 (頁 353)。實際上, 由 L^AT_EX 所產生之索引項目須重新整理之後, 再引入主檔案內, 但本例子檔案並未作此一處理。

主檔案第 1 行所引用的 `w-macro.ctx`, 包含所有的巨集套件與指令, 各指令之前有簡單的功能介紹。大體上, 各指令之定義方法及用法本書都有介紹, 故不再進一步說明。排版本例子書籍須執行下列步驟:


```

cwtex ecobook
latex ecobook
cwbibtex ecobook
cwidx ecobook
latex ecobook
latex ecobook

```

其中，第3道指令是以 `cwbibtex` 排版參考文獻。除了第4道指令之外，其餘各程式都可以在 WinEdt 視窗內直接執行。第4道指令是編排索引之處理動作，必須在 DOS 視窗內執行。

18.3 索引

除了小說、戲劇等文學作品之外，大部分的書籍都需要索引。清楚完整的索引是好書的必要條件。編製索引的工作非常繁瑣。以往電腦尚未應用於排版時，中文書很少有編製索引。`ETEX` 提供一些輔助工具，使編輯索引的工作簡化了許多。更重要的是，每一索引項都很精確。`ETEX` 的輔助工具無法處理中文，`cwTeX` 提供 `cwbibtex` 程式，以方便編製中文索引。

18.3.1 標註索引名詞

編輯索引的第一步驟是標註欲編入索引之名詞，使用之指令為 `\index`。譬如，在一篇討論資訊產品的書稿中，若「顯示器」一詞要編入索引中，則文稿中此一名詞出現之處，其後應鍵入 `\index{顯示器}`。此一指令完全不影響正文文稿內容之編排，其功能只是標註此一名詞應列入在索引中，使 `ETEX` 可以找出頁碼。

以上指令讓「顯示器」一詞在編排完畢之後單獨成為索引中之一項。如果要此一名詞排列在「週邊設備」索引項下，則我們應鍵入之指令為：

```
\index{週邊設備!顯示器}%
```

如果「週邊設備」一詞本身又是「電腦產品」底下的次項，則鍵入之指令再加一層，變成：

```
\index{電腦產品!週邊設備!顯示器}%
```

排版完畢之後，索引版面將如下例所示：

電視機, 102
 電腦產品, 25
 主機版, 37, 45
 週邊設備, 72
 滑鼠, 36, 63, 64
 顯示器, 35, 63
 電熱器, 135, 138

有時候, 索引項目本身須加上排版指令。譬如, 本書之索引中有 `\beta` 項目。輸入索引項時, 若直接鍵入 `\index{\verb+\beta+}`, 索引排序時, 此項目將排列於 `\verb` 處; 而正確的位置應該是在 `\baselineskip` 之後。遇有此種情形, 我們可以在索引項目之前加入一排序用文詞, 真正的索引項則置於其後, 兩者之間以 `@` 符號隔開。因此, 以上的索引項應輸入為:

```
\index{beta@\verb+\beta+}
```

索引名詞排序時, 將取用 `@` 字元之前的 `beta` 排序。

有些索引名詞是出現在連續的幾頁中, 欲標誌此類索引項, 可在索引名詞連續出現的開頭處使用下列指令:

```
\index{dvips@\textsf{dvips}|{}
```

本例中, 連續出現的索引名詞為 `dvips`, 指令 `|{` 標示開端。連續索引結束處則加上下列指令:

```
\index{dvips@\textsf{dvips}|})}
```

指令 `|)` 標示結尾。若開端是在第 12 頁, 結尾是在第 15 頁, 排版結果如下:

`dvips, 12–15`

由上可知, `\index` 指令內有一些符號具有特定用途。具體言之, 索引項目內若含有 `!`, `@`, 與 `|` 三個符號, 輸入在 `\index{...}` 指令內時, 其前面須加上 `"` 符號, 變成 `"!`, `"@`, 與 `"|`。進一步的細節請見 [Goossens et al. \(1994\)](#), 第 12 章。

`\index` 指令應與索引名詞愈靠近愈好。此外, 下指令時須小心不要讓排版結果多了不必要的空白。舉例來說, 在上一段文字開頭, 我們下了一個 `\index`

指令:

由上可知, `\index{}` 指令內有一些符號具有特定用途。

`\index{ 索引!特殊符號}`

`\index{ 特殊符號}`

具體言之, ...

其中, 每一個 `\index` 指令自占一行。排版之後, “具體言之” 之前可能會多一個空格。正確的下法是把 `\index` 置於第一行末端, 緊接在中文句點之後, 而且兩個 `\index` 指令中不要有空格。如果要自成一, 請在指令末端加上註銷指令:

由上可知, `\index{}` 指令內有一些符號具有特定用途。

`\index{ 索引!特殊符號}%`

`\index{ 特殊符號}%`

具體言之, ...

18.3.2 排版索引的步驟

索引是列出各名詞的頁碼所在, 因此我們應等到全書排版完成, 頁碼不再更動之後, 才開始編排索引。索引的編排必須經過下列幾個步驟:

1. 首先, 確定全篇書稿中那些名詞或文字要編入索引中, 並以 `\index` 指令標註出來。標註方法, 請見下一小節之說明。
2. 全文設定區須加上 `\usepackage{makeidx}`, 以引入 `makeidx` 巨集套件。同時, 指令之後須加上 `\makeindex` 一行指令。最後, 文稿末端欲排版索引處須加上 `\printindex` 指令。若不上此指令, `TeX` 仍然會進行索引編輯的工作, 但不會將索引列印出來。
3. 若文稿檔名為 `cxbook.ctx`, 以 `cxTeX` 轉換中文, 並且執行 `latex` 兩次之後, 硬碟中會出現輔助檔案 `cxbook.idx`。此檔案內含索引項目及其頁碼, 但尚未排序。每一索引項都置於 `\indexentry` 指令之後; 其內之中文字為 `TeX` 指令格式。
- 4a. 若為純英文檔案, 執行指令即可將各索引項正確排序:

```
c:\xtemp>makeindex cxbook
```

排序之後，硬碟內產生 `cxbook.ind` 檔案。接下來，再執行 `latex` 一次，文稿末端即自動排入索引項。

- 4b. 若是中文稿件，處理過程較複雜。為了避免錯誤，可使用 `cwidx.bat` 批次檔案，執行指令如下：

```
c:\xtemp>cwidx cxbook
```

請注意，勿鍵入延伸檔名，否則結果不正確。執行以上指令後，硬碟內將產生 `cxbook.ind`，最後，再執行 `latex` 一次，文稿末端即得索引。

以上簡單說明產生索引的過程，實際上排版時有些細節須特別處理，以下進一步說明之。首先，我們解釋 `cwidx.bat` 之內容。此批次檔計有 5 行指令，主要功能是在產生正確中文排序之索引檔案：

```
tex2xtc %1.idx
cwmkidx %1.xtc
copy %1.ind cwtemp.ind
cwtex -- + cwtemp.ind
copy cwtemp.tex %1.ind
```

為了讓中文索引正確排序，第 1 行指令先執行 `tex2xtc` 將 `cxbook.idx` 之 \TeX 指令形式之中文字指令轉回普通中文字，檔名自動取為 `cxbook.xtc`。

第 2 行指令執行 `cwmkidx` 程式，將檔案內之中英文索引項排序，檔名自動取為 `cxbook.ind`。`cwmkidx` 程式之功能與 `makeindex` 程式類似，但後者在處理某些中文字時會出現錯誤，原因是 `makeidx` 在處理索引時，將某些字元視為特殊字；而這幾個字元恰好是中文字所使用的內碼。因此，若某些中文字的內碼恰是這幾個特殊字元，排序時即會出現問題。

`cwTeX` 所提供的 `cwmkidx` 程式能避免以上這個問題。不過，中文字的排序是直接利用中文內碼之順序，因此排序結果並不會百分之百正確，原因如下：首先，中文 Big-5 內碼分常用字與非常用字兩部分。排序時，非常用字即使其筆劃較少，也會被排在常用字的後面。另外一種情況是使用者自行造字。自行造字之內碼通常是排於常用字與非常用字的後面，排序時即使是筆劃少也會排在最後面。因此，中文字排序之後有時候須略加調整才能得到正確的結果。欲調整中文排序，直接以文字編輯軟體修改 `cxbook.ind` 檔案即可。

除了排序之外, `cwmkidx` 或 `makeindex` 程式還會檢查文稿輸入之索引格式是否正確。`cwmkidx` 程式會將執行結果記錄於 `cwindexg.ilg` 檔案內, 其內容如下例所示:

```
Scanning input file ... (38 entries accepted, 0 rejected).
Sorting entries...done (193 comparisons).
Generating output file ... (87 lines written, 0 warnings).
Output written in cwindexg.ind.
Transcript written in cwindexg.ilg.
```

本例中, 第1行之“0 rejected” 訊息顯示原始索引檔案並無問題, 第3行所出現之“0 warnings” 訊息, 則說明排序後之索引檔也正確無誤。如果原先標誌索引名詞時發生錯誤, 訊息將記錄於 `cwindexg.ilg` 檔案內, 說明錯誤出現在那一行。底下的簡單例子說明錯誤是位於 `cwindexg.tmp` 檔案的第30行:

```
!! Input index error (file = cwindexg.tmp, line = 30)
```

事實上, `cwindexg.tmp` 是一暫存檔, 其來源是 `cxbook.xtc` 檔案。因此, 我們應至 `cxbook.xtc` 檔案的第30行更正錯誤。除了標誌索引項目可能發生錯誤之外, 若索引名詞內容太過複雜, `tex2xtc` 程式將 `cxbook.idx` 轉換為 `cxbook.xtc` 時也可能產生錯誤。

更正所有錯誤之後, 即可作下一階段的處理。`cwmkidx` 程式雖然產生正確排序的索引檔案, 但檔案內之中文字並非 \TeX 指令格式。因此, 排版之前須先轉換索引檔案內之中文字。`cwidx.bat` 程式後三行指令即透過一暫存檔案作中文轉換之動作, 最後之檔案仍然命名為 `cxbook.ind`。

以上的步驟可以在文稿尾端產生排版索引, 但其格式是 \LaTeX 標準的兩欄格式。若版面寬夠大, 我們也可以三欄排版。此外, 在中文書籍的索引中, 我們通常在索引版面上加入筆劃數之小標題以方便查詢。中文索引項目排序之後, 相同筆劃之索引項會集中在一起。依慣例, 在名詞筆劃數目為10之索引項之前, 通常會加上「十劃」之小標題; 筆劃數為11之索引項之前, 則加上「十一劃」小標題。加上小標題的方法很簡單, 以下是一個簡單的例子。在 `cxbook.ind` 中, 每一索引項之前都有 `\item`, `\subitem` 或 `\indexspace` 等指令。此為 \LaTeX 所自動加上者, 前兩項指令控制索引之排版形式, 後一項指令控制間距。找到正確筆劃數之索引項目, 在其之前加入下列指令:

```

\item ...
\par\medskip
\noindent{\ctxfbb 十劃}
\par\medskip
\item ...

```

排版後，小標題將以粗黑字體居左編排，上下各空出 `\medskip` 之空白。

排版之後，索引首頁會加上 **Index** 英文字為標題。如果是中文稿，而希望以粗黑體「索引」作為標題，可在全文設定區重新定義如下：

```

\renewcommand{\indexname}{\ctxfbb 索引}

```

18.4 裁切記號

排版書籍時，我們可在每一頁版面上下左右四個角落加上裁切記號 (cropmarks)，書籍印刷之後，裝訂廠即可依此記號裁切紙張。欲畫出裁切記號，我們可使用 `crop` 巨集套件，作者為 Melchior Franz。底下是裁切記號的指令例子。

```

\documentclass[12pt]{book}
\usepackage[cam]{crop}
\paperheight=22.5cm
\paperwidth=15.5cm
\setlength{\textheight}{17.5cm}
\setlength{\textwidth}{11cm}

```

引用巨集套件時，我們加上 `cam` 選項，此選項選擇裁切記號為圓圈加上十字符號，這是排版界的標準記號。若改用 `cross` 選項，裁切記號將變成十字型。

`crop` 巨集套件是依據紙張尺寸與正文方塊尺寸排版裁切記號。本例子紙張尺寸為 22.5×15.5 公分；正文方塊為 17.5×11 公分。若使用 `geometry` 巨集套件，則紙張與正文方塊大小都已定義於其內，不須再另行定義。請參考 18.2 節 (頁 349) 所介紹的排版書籍的例子。

排版時，`crop` 巨集套件會在版面上方加上檔名、排版日期、頁碼等資訊。但是新版 `crop` 在處理中文稿時有時候會出現亂碼。為避免此一問題，你可以用舊版 `crop.sty` 替代新版。舊版檔案置於光碟 `\support` 檔案夾內，新版則安裝於 `c:\texmf\tex\latex\crop` 檔案夾。

19 網路出版

對許多人來說,上網取用資訊已經是生活中不可或缺的一項活動。因此,愈來愈多的人想要學習如何將資訊發表在網路上。網路出版的概念是在 $\text{T}_{\text{E}}\text{X}$ 系統之後才出現的, $\text{T}_{\text{E}}\text{X}$ 系統當然不可能包含網路出版的功能。不過,不少 $\text{T}_{\text{E}}\text{X}$ 使用者希望能將排版結果送上網路發表。譬如說,學術論文若能置於網路上流通,對於作者與讀者都可省下許多郵件往來的成本。同樣的,越來越多的老師將授課的講義、習題或者學期成績,直接置於網路上,由學生自行讀取。本章的目的是從 $\text{E}_{\text{T}}\text{X}$ 系統的角度說明如何作網路出版 (web publishing)。有關於 $\text{E}_{\text{T}}\text{X}$ 應用於 WEB 上排版工具之說明,請見 [Goossens and Rahtz \(1999\)](#)。

網路排版最常使用的語言是 HTML (Hypertext Markup Language)。我們進入網站後,呈現在顯示器上的圖形或文字,大部份是利用 HTML 語言所建構出來的。不過,也有不少網頁資訊是使用 PDF (Portable Document Format) 格式。HTML 與 PDF 各有其相對長處與弱點,本章的目的不是要探討這兩種語言的差異,而是要介紹幾套有用的工具程式與巨集套件,透過這些工具,即可將 $\text{T}_{\text{E}}\text{X}$ 排版結果轉換成 PDF 或 HTML,送上網路發表。以下的說明假設使用者對於 HTML 語言已有基本的了解。

19.1 HTML 與 PDF 的比較

介紹轉換工具程式之前,我們首先簡單比較 PDF 與 HTML 之特點。PDF 檔案格式是由 Adobe 公司所發展,事實上它是 PostScript 語言的簡化與延伸。所謂簡化,是指 PDF 並無 PostScript 某些程式語言功能。但反過來說,Adobe 公司在 PDF 中加入一些網路排版的功能,例如填表 (form) 功能,因此它是 PostScript 語言的延伸。

PDF 檔案格式通常用於排版專業文稿。相對而言,HTML 則適用於版面不太複雜的文稿。舉例言之,若文稿中有複雜的數學式,HTML 語言難以應付;PDF 則

輕而易舉。雖然 PDF 有相當強的功能可以作複雜的網路排版,但這也表示自行下指令編排並不容易。幸運的是, T_EX 使用者不須再學習此一語言。我們可以使用現成的工具程式將 T_EX 的排版結果直接轉換成 PDF。透過這些工具, PDF 可以說得來全不費工夫。

對於 T_EX 的使用者而言, PDF 雖然容易轉換,但是它的主要問題之一是檔案太大。特別是中文稿件中若使用許多字體種類,一份30頁的稿件,檔案可能高達 600K 大小。如果文稿內容不大複雜,而且網路傳輸速度是重要考慮,則 HTML 語言是較佳的選擇。HTML 語言的概念與 T_EX 語言相當類似。從使用者的角度來看,我們甚至可以說 HTML 是一套簡化的 T_EX 排版系統。欲排版 HTML 文稿,我們使用任意的文字編輯軟體輸入文字與排版指令。送上網站之後,利用瀏覽器 (browser) 即可觀看排版結果,必要時可以列印出來。因此,瀏覽器就是 HTML 系統的預視與列印軟體。

就一般性質的排版功能而言, HTML 語言比 T_EX 簡單多了。不過,就網路排版而言, HTML 有一些功能是 T_EX 所沒有的。譬如, HTML 可以排版填表 (form); 當出現在畫面上時,使用者填入相關資料之後,程式可以進一步處理。網路購物時,我們須填入姓名、地址、信用卡帳號,這就是利用 HTML 的填表功能。

學習 HTML 語言並不困難,但如果為了偶而一次的網路排版而花時間去學習,也是不小的負擔。所幸的是,網路上也有幾套免費下載的工具程式,可以直接將 T_EX 排版結果轉換為 HTML。本章將介紹的是 \LaTeX 2HTML 程式,其功相當強,而且能夠處理中文。

綜合以上所述, PDF 與 HTML 是網路排版的兩套語言。透過以下所介紹的工具程式,使用者可以將排版結果直接轉換為其中之一。因此,剩下來的問題是要選擇哪一種格式作為網路排版? 這個問題就必須由你自行決定了! 底下首先介紹轉換為 PDF 之工具程式。

19.2 轉換為 PDF 格式

將 T_EX 排版結果轉換為 PDF 有幾個可行的路徑。不過,轉換的結果品質有好有壞。使用 PDF 格式的主要考慮是排版品質; 其中一個重要關鍵則是使用之字型規格。如果要求最佳品質,不管你用的是那一種工具軟體轉換,都須設定使用描邊字型。若使用描點字型,則在顯示器上或是列印結果,字跡都會模模糊糊的。

在介紹轉換工具之前，我們先簡單說明如何讀取網路上的 PDF 檔案。欲讀取 PDF 檔案，須使用 Acrobat Reader 或 GSview。前者是 Adobe 公司的免費使用軟體，後者是 Ghostscript 在 Windows 系統上的介面軟體，作者為 Russell Lang。Acrobat Reader 的優點是它所呈現的 PDF 畫面品質較精細；另外，它與網路瀏覽器有較佳的結合。若在較新版的瀏覽器下安裝 Acrobat Reader，以滑鼠點選網路上的 PDF 檔案時，瀏覽器會自動執行 Acrobat Reader 軟體，在瀏覽器的視窗內顯示檔案內容。反之，若使用 GSview，我們須先將 PDF 檔案下載、儲存於硬碟中，才能預覽、列印。不過，GSview 的優點是它有多方面的功能。譬如，除了 PDF 外，它也可以預覽 PostScript 檔案，甚至也可以將 PostScript 檔案轉換為 PDF。

以下依序介紹使用 pdf \LaTeX ，Ghostscript，及 Acrobat Distiller 軟體轉換 PDF 的方法。如果你僅是要將排版結果送上網站供他人下載，則這些工具程式大抵而言已能夠滿足需求。但是，如果你希望 PDF 文稿也具有網路文件之特性，例如串接文稿 (hypertext) 或者填表 (forms)，最簡單的方法是在 \LaTeX 文稿中引用 hyperref 巨集套件。此巨集套件可以和上述的工具程式配合使用，我們將在最後作簡單介紹。

19.2.1 Ghostscript

除了 Distiller 程式外，Ghostscript 也提供工具程式可將 PostScript 檔案轉換為 PDF。若利用較新版 Ghostscript 作轉換，PDF 檔案之品質甚佳。而且，文稿內所引用之 EPS 圖形檔案也能正確轉換出來。

在 WinEdt 視窗上方設定有一 ps2pdf 圖像 (見圖 3.3, 頁 43)，文稿排版完成，轉換為 PostScript 格式後，點選此一圖像，即可開啓 Ghostscript 工具程式，將排版結果轉換 PDF 檔案。

19.2.2 pdf \LaTeX

以上所介紹的是如何將 PostScript 檔案轉換為 PDF，pdf \LaTeX 程式則更進一步，它可以將 \TeX 或 \LaTeX 文稿直接排版為 PDF。此程式延伸 \TeX 系統的功能，這是 Sebastian Rahtz 與 Hàn Thê Thành 所合力創作。

相對於 Distiller 而言，pdf \LaTeX 所轉換出來的 PDF 檔案較大。另外，如果文稿內引用了外製圖形，這些圖形必須是 PDF 或 JPEG 格式；PostScript 圖形不能

直接使用。雖然有這些限制,如果你只是偶而轉換 PDF 檔案, pdf \LaTeX 程式不失為是一個好的選擇。事實上,它有一項特點是其他工具程式所沒有的: 可以使用 True Type 字型排版。詳情請見該軟體之說明檔。

19.2.3 Acrobat Distiller

Acrobat Reader 是 Adobe 公司的產品,可自網路免費下載。Adobe 是商業軟體公司,它的生存靠的是出售軟體賺取利潤。既然如此,為何它會免費提供 PDF 預視軟體呢? 原因很簡單,它出售 Acrobat 軟體套件,其中最重要的就是製作 PDF 之工具軟體: Distiller。此一工具軟體的主要功能是將 PostScript 檔案轉換為 PDF。因此,如果 \TeX 排版結果已經由 DVIPS 轉換為 PostScript 格式,只要再執行 Distiller 一次,即可得到 PDF,非常方便。

比起底下兩小節所介紹的 pdf \LaTeX , Distiller 有兩項特點。第一,它所轉換出來的檔案較小; 第二,原先 PostScript 檔案內所引用之外製圖形,也可以成功轉換。不過,如前所述,決定 PDF 文件之品質的關鍵因素是在於 PostScript 檔案所使用的字型規格。以 DVIPS 產生 PostScript 檔案時,如果使用的是描點字型,則轉換後之 PDF 文件不管是預視或列印,品質都不理想。欲求良好品質,PostScript 檔案須設定使用描邊字型。

19.2.4 hyperref 巨集套件

為方便交叉引述 (cross-referencing), \LaTeX 提供 `\label`, `\ref`, 與 `\pageref` 三道指令,網路排版的 HTML 與 PDF 語言在此一方面有更強的功能。譬如,我們可以由文稿某處跳到另一處; 可以由甲文稿跳到乙文稿; 甚至可以由甲文稿跳至某一網站,或者發電子信函給某人。為方便說明,我們稱此為超連結 (hypertext links)。

如果希望轉換後之 PDF 文件具有網路文件之超連結功能,方法之一是使用 Sebastian Rahtz 所寫的 hyperref 巨集套件。首先,在全文設定區以下列指令引用巨集套件:

```
\usepackage[dvips]{hyperref}
```

方括號之 dvips 選項表示接下來要以 DVIPS 與 Distiller 程式轉換為 PDF。若使用 pdf \LaTeX 轉換,則選項應為 pdftex。

舉例言之,若文稿中之表格標題以 `\caption` 指令排版,而且以 `\label` 指令設定徵引標的;以 `\ref` 指令設定徵引功能。排版之後,假設此一表格為文稿中第 3 表格,則文稿中徵引處自動排版為「表 3」。以上是 \LaTeX 本身即具備的功能。引用 `hyperref` 巨集套件之後,原 \LaTeX 之引述指令在 PDF 檔案中將具有超連結能力。具體言之,PDF 檔案之表格標題與檔案內之「表 3」處將顯示為特別顏色。以 Acrobat Reader 預覽時,若以滑鼠點選文稿內之「表 3」,畫面即跳至表 3 處。以上的例子說明,若使用 `hyperref` 巨集套件,PDF 檔案內之徵引功能可以說得來全不費工夫。

除了圖表之外,章節目次、註解、數式編號也會出現徵引功能。如果進一步希望文稿有 HTML 語言特有之超連結功能,須使用 `hyperref` 特別提供之指令。譬如,由文稿某處要連結特定網址,可使用 `\href` :

```
\usepackage[pdfTeX,colorlinks=true]{hyperref}
... 參見\href{http://ceiba.cc.ntu.edu.tw/tmwu}{cwTeX 網頁}。
```

以 `pdf \LaTeX` 排版之後,PDF 檔案內將出現“...參見 cwTeX 網頁”,其中“cwTeX 網頁”為紅色。以 Acrobat Reader 讀 PDF 檔案時,以滑鼠點選紅色,即可聯結到 cwTeX 網頁。如果你是要以 DVIPS 將排版結果先轉換為 PostScript 檔,再用 `distiller` 或 `Ghostscript` 轉換為 PDF,則引用 `hyperref` 巨集指令時應選用 `dvips`, 而非 `pdftex`。

文稿內之甲文字串欲連結至乙文字串,可以使用 `\hyperlink` 與 `\hypertarget` 指令。例如,文稿甲處有下列文句:

```
...
有關於排版問題,參見\hyperlink{there}{ 下文}之說明。
```

其中,“there”為自行選定之標誌。文稿乙處以 `\hypertarget` 定義其位置:

```
...
\hypertarget{there}
本研究成果之排版 ...
```

排版之後,以 Acrobat Reader 預覽,PDF 檔案內“下文”兩字為紅色,以滑鼠點選,版面即跳至文稿 `\hypertarget{there}` 指令處。詳細說明請參考 `hyperref` 巨集

套件之說明檔。

以 Acrobat Reader 預覽 PDF 檔案時，畫面左邊另有一欄，存放 PDF 檔案之章節目錄，這就是所謂的 bookmarks (書籤)。PDF 檔案之 bookmarks 功能類似普通文稿的章節目錄。若文稿分 5 節，以滑鼠點選 bookmarks 內第 3 節，畫面立即顯示第 3 節開頭之內容。對於長篇文稿而言，bookmarks 有其方便之處。hyperref 巨集套件處理 \LaTeX 文稿時會將各章節之標題匯總為 bookmarks。但若是中文文稿，bookmarks 顯示的卻是一些奇怪的指令。事實上，這的確是章節標題，不過卻是中文標題經 cwTeX 轉換後之指令。如此的內容對於使用者並無任何幫助。

據我們所知，要在中文 PDF 上製作出正確的書籤，唯一的辦法是以 Adobe Acrobat 軟體直接編輯轉換出來的 PDF 檔案。

19.3 \LaTeX 2HTML

網路排版第二條途徑是 HTML。事實上，HTML 應該是網路排版的主要語言。若版面不太複雜，只要對 HTML 語言有簡單的了解，自行以文字編輯軟體寫出網頁並不困難。不過，若文稿含有數式或圖形，網頁的製作就困難一些，原因是目前 HTML 語言處理數式的能力不強。

網路上有好幾套工具程式可以轉換 \LaTeX 文稿變成 HTML，不同程式處理數式的方法也不同。本節所要介紹的是 \LaTeX 2HTML，此一程式是由多人合作完成，原始作者為 Nikos Drakos。此一工具是由 perl 語言寫成，實際執行時須用到 DVIPS, Ghostscript 及其他工具程式。這是一套複雜的系統，但也可預期其功能甚強。 \LaTeX 2HTML 主要特點是它將所有的數式都轉換為描點圖形，而且所有的轉換動作都自動處理，不勞使用者費心。

對於中文稿而言， cwTeX 提供簡單的指令及工具程式，讓使用者可以設定文稿內較大的中文標題以描點圖形排版，文稿內之中文字則直接輸出，不作轉換。以下將以幾個例子說明排版方法。

19.3.1 安裝

執行 \LaTeX 2HTML 時，除了須使用 \LaTeX 與 Ghostscript 之外，還須使用 perl 與 netpbm 兩組程式。安裝 \LaTeX 2HTML 之過程較複雜，故我們提供一簡單的安裝

程式 `setup12h.bat` 。萬一此程式在你的系統無法正確運作, 你還可以依照下文所描述步驟手動安裝。

● 程式安裝

以程式安裝係指使用 `cwTeX` 所提供的 `setup12h.bat` 程式安裝, 步驟如下:

1. 請確定電腦中之 Ghostscript, MiKTeX 與 `cwTeX` 已能順利執行。
2. 進入光碟 `\latex2html` 檔案夾內, 將 `perl-win32.zip` 與 `netpbm.zip` 解壓縮於 `c:\` 之根目錄。第 1 個檔案含 `perl` 程式, 第 2 個檔案含 `netpbm` 程式集。解壓縮後, 請在 `c:\autoexec.bat` 檔案最後請加入底下一行搜尋路徑設定指令:

```
path=c:\perl\bin;c:\netpbm\bin;%path%
```

重新開機, 讓設定生效。

3. 進入 DOS 視窗, 再進入光碟 `\latex2html` 檔案夾內, 直接執行安裝程式:

```
d:\latex2html>setup12h
```

如果你的作業系統是 Win2000 或 Windows XP, 執行程式時可能會在底下的訊息之後停頓下來:

```
checking dvips version ...
```

請等候約 1 分鐘之後, 按 [Enter] 鍵, 再稍候一會兒, 程式會繼續執行, 並完成安裝。

欲了解是否正確安裝, 請依下一小節之說明, 編排 `test-h1.tex` 測試檔。若轉換後之 HTML 檔案內有圖形出現, 即表示安裝成功。若不成功, 請依底下之手動安裝步驟再試一次。

● 手動安裝

若以上之自動安裝無法成功, 請試依底下之步驟安裝。前兩個步驟與自動安裝之過程相同, 如果你已經執行過了, 請由第 3 步驟開始。

1. 請確定電腦中之 Ghostscript, MiKTeX 與 cwTeX 已能順利執行。
2. 進入光碟 \latex2html 檔案夾內, 將 perl-win32.zip 與 netpbm.zip 解壓縮於 c:\ 之根目錄。第 1 個檔案含 perl 程式, 第 2 個檔案含 netpbm 程式集。解壓縮後, 請在 c:\autoexec.bat 檔案最後請加入底下一行搜尋路徑設定指令:

```
path=c:\perl\bin;c:\netpbm\bin;%path%
```

重新開機, 讓設定生效。

3. 將光碟內之 12h2k-1.zip 解壓縮於 c:\xtemp 檔案夾內。進入 c:\xtemp 檔案夾, 以文字編輯軟體開啓 prefs.pm 。第 129 行應改為:

```
$prefs{ 'PREFIX' } = 'c:\\latex2html';
```

請注意, 檔案夾名稱是以連續兩個反斜線 \\ 起頭。以上設定指示將程式安裝於 c:\latex2html 。為簡化文字說明, 以下僅列出其他應修改之行數, 請將各行 '...' 內之文字改為以下所列:

```
196 行: 'c:\\texmf\\tex\\latex\\latex2html';
```

```
244 行: 'c:\\gstoools\\gs5.50\\gswin32c';
```

第 196 行之設定是假設你使用 MiKTeX, 且系統安裝於 c:\texmf 。第 244 行設定 Ghostscript 5.50 版程式之路徑。若你安裝的是 6.52 版, 應修改為:

```
'c:\\gs\\gs6.52\\bin\\gswin32c';
```

以上修改完成之後, 請進入 c:\xtemp 檔案夾, 執行下列指令:

```
c:\xtemp>config
```

如果你的作業系統是 Win2000 或 Windows XP, 執行程式時可能會在底下的訊息之後停頓下來:

```
checking dvips version ...
```

請等候約1分鐘之後, 按 [Enter] 鍵, 再稍候一會兒, 程式會繼續執行。為了解以上步驟是否正確, 請以文字編輯軟體開啓 config.log 檔案, 若程式正確執行, 其中應有一行文字:

```
checking dvips version ... 5.86
```

若無此行文字, 表示安裝程式無法辨識 dvips 版本, 以上步驟須重新來過。

4. 接下來, 請再以文字編輯軟體開啓 c:\xtemp 內之 l2hconf.pm , 第 52 行請改為:

```
@IMAGE_TYPES = qw/gif png);
```

此外, 第 151 行原為:

```
# $DVIPSOPT .= " -E";
```

請將行首之 # 刪除。以上修改完成之後, 進入 c:\xtemp 檔案夾內, 執行:

```
c:\xtemp>install
```

安裝程式會將 \TeX 2HTML 檔案複製於 c:\latex2html 檔案夾內。

5. 以上安裝步驟若一切順利, c:\latex2html\bin 檔案夾內會產生一個執行檔 latex2html.bat 。為方便起見, 請將此檔案複製一份於 c:\texmf\cwtex 檔案夾內。
6. 最後, 為了正確中文字, 請將 c:\texmf\cwtex\util 檔案夾內之 cwtex.perl 檔案複製於 c:\latex2html\styles 檔案夾內。

19.3.2 使用方法

\TeX 2HTML 系統甚為複雜, 但從使用者的角度來看, 排版一般性文稿並不困難。完整的說明檔置於 \latex2html\docs 檔案夾內, 請自行參考。說明使用方法之前, 我們簡單說明此一系統之設計理念。HTML 語言可以處理簡單的數學式, 但太複雜的數式就無能為力。不過, HTML 處理圖形的能力不錯, 因此可能的方法之一是將數式轉為圖形。 \TeX 2HTML 轉換數式為圖形的方法如下: 先將文稿排

版一次,找出其中的數式或引用之外製圖形;將這些數式與圖形利用 DVIPS 轉換為一個一個的 PostScript 檔案。接下來,利用 Ghostscript 程式將檔案轉換為 GIF 格式。最後,再把原文稿轉換為 HTML 格式,並將所有的數式或圖形以 GIF 圖形檔替代。以上的轉換步驟全部自動處理,不勞使用者費心。

cwTeX 例子檔案夾內有5個測試檔, `test-h1.tex`, ..., `test-h5.ctx`。為方便測試,我們建議建立一子目錄: `c:\tex\html`, 將5個測試檔複製至子目錄內。第一個檔案 `test-h1.tex` 為純英文檔。此測試文稿內含一圖形,檔名為 `dmark.eps`。欲轉換為 HTML 檔案,先以 `latex` 編排 `test-h1` 兩次之後,執行 `latex2html`:

```
c:\tex\html>latex test-h1
c:\tex\html>latex test-h1
c:\tex\html>latex2html test-h1
```

`latex2html.bat` 為一批次檔,置於 `c:\texmf\cwtex` 檔案夾內。批次檔中設定一些選項,若這些選項不合你的需求,請自行修改之。

若執行成功, `c:\tex\html` 檔案夾之下將出現 `\test-h1` 次檔案夾,HTML 即置於其中。以瀏覽器開啓 `index.htm` 或者 `test-h1.htm` 檔案,即可看到結果。檔案夾內可發現約二十來個 GIF 檔案,這些都是由文稿內之數式與引用之外製圖形檔轉換而來。

實際轉換之前為何須先執行 `latex` 兩次? 原因是文稿中若使用了超連結 (`hyperlink`) 指令,則 `TeX2HTML` 轉換時須讀取輔助檔案 `test-h1.aux` 內之資訊。先執行 `latex` 兩次,其目的即在產生輔助檔案。因此,若文稿並未使用 `TeX` 的引述指令,如 `\ref` 或 `\label`;也未使用 `TeX2HTML` 的超連結指令,如 `\htmlref`,則文稿可直接轉換,不需先前的步驟。換言之,某一文稿若執行 `latex` 時硬碟中不產生 `.aux` 輔助檔案,文稿可直接執行 `latex2html.bat` 程式轉換。反之,若文稿中使用了超連結指令,但轉換之前未曾先產生 `.aux` 輔助檔案,則產生之 HTML 檔案就喪失了超連結功能。

以上測試檔名為 `test-h1.tex`,但轉換之後會產生兩個 HTML 檔案,分別取名為 `test-h1.htm` 與 `index.htm`。這兩個檔案內容相同。在 HTML 語言中,首頁 (`main page`) 須取名為 `index.html`,這是 `TeX2HTML` 產生兩個檔案的原因。如果你是要設計網頁之首頁,原始 `TeX` 檔案應直接取名為 `index.tex` 或 `index.ctx`。

19.3.3 轉換中文稿

$\text{\TeX}2\text{HTML}$ 如何處理中文呢？中文可以用兩種方式處理：第一是把中文視為圖形；第二種方式是中文不作任何加工，直接轉入 HTML 檔案中。要把中文視同圖形處理，方法很簡單，僅須在 \TeX 文稿中讓中文以數式模式出現即可。譬如， c\TeX 網頁是以 $\text{\TeX}2\text{HTML}$ 製作，其中一項標題是「總體經濟學」。在文稿中，此一標題的文字與指令如下：

$$\$\mbox{\Large \ctxfbb 總體經濟學}}\$$$

$\text{\TeX}2\text{HTML}$ 認定此為數學式，即將這 5 個中文字轉換為一個圖形檔。此例中，5 個中文字視為一圖形，故轉換結果僅得一圖形檔案。要把 5 個中文字轉換為 5 個檔案也可以，細節請參考說明檔。以上是中文標題為例，英文標題當然也可以使用同樣方式處理。

第二種處理中文的方式是不作任何加工，直接移入 HTML 文稿內。不過， $\text{\TeX}2\text{HTML}$ 系統設計是以英文為主要對象；文稿中若有中文字，直接處理之後會變得無法辨識。為了正確處理中文， c\TeX 提供一巨集套件 cwtex.sty 及一對應之 perl 程式 cwtex.perl 。任何中文稿若要直接納入 HTML 檔案內，請在全文設定區加入下列指令：

$$\usepackage{cwtex}$$

在作普通文字排版時，此一巨集套件並無任何作用。但是，在作 HTML 轉換時， $\text{\TeX}2\text{HTML}$ 會自動引入 cwtex.perl 程式，讓中文字得以正確處理。

事實上，以上處理 cwtex 巨集套件的方法是 $\text{\TeX}2\text{HTML}$ 的標準程序：若 \TeX 文稿引用了任何巨集套件，系統中必須有一對應的 perl 檔案，否則不能正確處理。 $\text{\TeX}2\text{HTML}$ 說明檔內列出它目前所支援的巨集套件，未來支援巨集套件會越來越多。不過，文稿若要轉換為 HTML，最好僅使用標準的 \TeX 指令。欲使用額外的巨集套件，請先確定它是在 $\text{\TeX}2\text{HTML}$ 的支援範圍之內。

另一個問題是字體指令。HTML 語言僅提供簡單的字體指令，如正體、斜體、放大、縮小等；但無法使用特別的字體，如 Garamond。¹ 就英文字部分而言， $\text{\TeX}2\text{HTML}$ 無法處理 \fontsize 指令，因此也無法處理類似 \sz12 之巨集指令。

¹如果要使用特別字體作網路排版，應使用 PDF 格式，而非 HTML。

就中文字而言，特定的標題或句子若要轉換為圖形檔，可加入中文字體指令；要直接移入 HTML 文稿內之中文則不應使用中文字體指令，請見以下之說明。

第 2 個測試檔 `test-h2.ctx` 為中文稿，該檔案前端簡單說明執行步驟。此檔案之內容單純，無數學式或圖形，但文稿中使用超連結指令 `\htmlref` 與 `\label`。因此我們首先將此檔案依一般 \TeX 文稿處理：先執行 `cwtx`，再執行 `latex` 兩次。排版完畢，硬碟中出現 `test-h2.aux` 輔助檔案，這是下一階段所需要的。

接下來，須再執行 `cwtx` 一次，但指令中須加入 `-i` 選項：

```
c:\tex\html>cwtx -i test-h2
```

此選項之作用為何？以文字編輯軟體開啓 `test-h2.ctx`，第 19 行為 `\ctxfoff`。此一指令的作用是要求 `cwtx` 程式不要將以下的中文字轉換為 \TeX 指令。與 `\ctxfoff` 作用相反的是 `\ctxfon`，指示轉換以下之中文字。利用這兩個指令，執行 `cwtx` 程式時，我們可以控制將 `.ctx` 文稿中某段中文照原樣轉錄於 `.tex` 檔案內，不作轉換。再利用上面所說的方法， $\text{\TeX}2\text{HTML}$ 程式可將中文字直接置於 HTML 檔案內。不過，`\ctxfoff` 與 `\ctxfon` 指令要產生作用，執行 `cwtx` 時須加入選項 `-i`。若不加入選項，這兩個指令毫無作用。

上面已經說明， $\text{\TeX}2\text{HTML}$ 系統是針對英文設計。為了讓它正確處理中文，我們須在中文稿的全文設定區加入巨集套件：

```
\usepackage{cwtx}
```

文稿中加上此巨集套件時， $\text{\TeX}2\text{HTML}$ 處理時將自硬碟中尋找 `cwtx.perl` 檔案，其中包含有正確處理中文之指令。基本上，所有的中文稿都須使用此巨集套件，否則中文字無法正確處理。

綜合以上所述，欲轉換出 HTML 檔案，共須執行 5 道指令：

```
deltree %1
cwtx %1
latex %1
latex %1
cwtx -i %1
latex2html %1
```

若經常作轉換，我們可以寫一簡單批次語言，一方面節省時間，同時也避免錯誤。

我們將以上5道指令集合為 `new.bat` ; 欲轉換 `test-h2.ctx` 檔案, 指令如下:

```
c:\tex\html>new test-h2
```

如上所述, $\text{\LaTeX}2\text{HTML}$ 所產生之檔案全部置於 `\test-h2` 子目錄內。如果我們修改了 `test-h2.ctx` 文稿, 執行轉換指令之前應先將子目錄內檔案刪除, 這是 `new.bat` 批次檔第1行指令的作用。

第3個例子 `test-h3.ctx` 內容與上一例子相同, 但標題改為圖形檔, 故可選用較大之字體。因為標題內有中文字, 故 `\ctxfoff` 指令置於中文標題之後。(為什麼?) 標題之圖形檔將自動命名為 `img1.gif`。排版完成, 上載 HTML 時, 請記得將圖形檔也上載。若文稿中有兩個圖形檔, 第2個圖形檔將命名為 `img2.gif`。

現假設文稿內稍後另有一行標題, 也是要以圖形方式處理, 則在第二個標題之前須加入 `\ctxfon`, 啟動中文轉換功能以轉換第二行標題。但是, 請記得在中文標題的下一行加入 `\ctxfoff` 指令關閉之。否則, 以下所有的中文字會一個一個全部轉換成圖形。

第4個例子延續上述例子, 但文稿內有一表格; 表格標題是以 `\caption` 指令排版, 以 `\label` 指令標誌, 文稿他處則以 `\ref` 指令徵引表格。排版時, 此表格自動編為「表1」。為產生引述功能, \LaTeX 排版時會產生 `test-h4.aux` 輔助檔案, 其中與表格有關的一行內存有表格標題。不過, 標題內之中文字已轉換為 $\text{T}_{\text{E}}\text{X}$ 指令。 $\text{\LaTeX}2\text{HTML}$ 若直接讀取此一行指令, 因無法辨識指令內容, 會產生錯誤訊息。故轉換之指令較複雜, 依序如下:

```
cwtex -f test-h4
latex test-h4
latex test-h4
tex2xtc test-h4.aux
copy test-h4.xtc test-h4.aux
cwtex -i test-h4
latex2html test-h4
```

請注意, 為配合第4行之 `tex2xtc` 轉換程式, 第1行指令執行 `cwtex` 時須加上 `-f` 選項, 指示使用舊式字體指令。第4行指令是利用 `tex2xtc` 程式將 `test-h4.aux` 檔案內之 $\text{T}_{\text{E}}\text{X}$ 指令轉換回中文, 轉換後之檔名自動取為 `test-h4.xtc`。第6行指令再將此檔案覆蓋原有之 `test-h4.aux`。

第5個例子為 `test-h5.ctx` 檔案,這是製作 `cwTeX` 網頁之原始檔案之簡化版本。原首頁之照片以一張貓的圖畫替代。執行成功之後,首頁有兩個超連結點,一是連往「總體經濟學」,另一連往「`cwTeX` 排版系統」。

以上的例子使用了 `TeX2HTML` 程式所特別提供的指令,如 `\htmlref` , `\label` , `\htmladdnormallink` 等,這些指令讓文稿具有超連結功能,可從本文稿連結至其他檔案、網站、或傳送電子郵件等。使用方法請見 `TeX2HTML` 說明檔案。

20 造字

中文排版的問題是：中文字永遠有不足的問題。標準的中文 TrueType 字型檔內含 13,053 個中文字。但有時候，我們會碰到一些特別的字，譬如，清末台灣的製糖所稱為「糖廊」，其中「廊」並不在標準的 TrueType 字型檔內。萬一用到字型檔沒有的中文字，使用者只好自行造字。此外，`cwTeX` 系統雖然提供 23 套中文字體，但你希望能使用 TrueType 其他字體，此時，我們可以將 TrueType 字型檔轉換為 `cwTeX` 格式。

造字時，我們可以造描邊字 (outline font)，也可以造描點字。描邊字的使用彈性大很多，但需要有特別的軟體才能造出。本章底下將簡單說明造描邊字的過程。在 Windows 作業系統上，我們可以找到許多字型廠商所生產的字型；但這些字型無法直接用於 `cwTeX` 系統上。`cwTeX` 系統發展小組的翁鴻翎先生費心寫了兩套轉換軟體，利用這兩套軟體即可將大部分的 Windows 字型檔轉換為 `cwTeX` 系統之格式。本章主要說明這兩套軟體的使用方法。

20.1 轉換 TrueType 字型檔為 `cwTeX` 字型檔

如果你有喜好的 TrueType 中文字型檔，利用本節所介紹之軟體即可將之轉換為 `cwTeX` 系統之字型格式。我們首先須聲明的是，中文字型檔和一般電腦軟體一樣，都受到相關的智慧財產權的保障。每一家字型軟體公司對於其字型軟體之使用有不同的規範，在轉換字型軟體之前，請先了解字型公司之使用條件。我們雖然提供字型轉換工具程式，但使用此軟體所可能涉及的法律問題，概與作者無關。由於 TrueType 的字型資料結構複雜，為了分析的方便，我們就權衡性的把 TrueType 字形概分為傳統的 TrueType 字形和 unicode 字形。

20.1.1 `cwpt1` 與 `cwttf` 軟體

字型轉換工具有兩套，分別名為 `cwpt1` 與 `cwttf`。第一套軟體 `cwttf` 的源頭是

`ttf2pfb` 程式，作者是李君宇先生。原程式是配合 CJK 系統，但只能處理傳統的 TrueType 字型，而無法處理較新的 unicode 字型。所謂 CJK 系統是一套可以處理中文 (Chinese)，日文 (Japanese)，與韓文 (Korean) 之 $\text{T}_{\text{E}}\text{X}$ 延伸系統，故名爲 CJK 系統，作者爲 Werner Lemberg。翁鴻翎修改並加強 `ttf2pfb` 程式之功能，使之可以轉換出 `cwTEX` 系統格式之字型檔，並且可處理傳統的及 unicode 之字型檔。

第二套軟體 `cwttf` 的源頭是 `ttf2pt1` 程式，作者是 Mark Heath，是由 Andrew Week 的 `ttf2pfa` 修改而來，現在主要的維護者是 Sergey Babkin。翁鴻翎修改並加強 `ttf2pfb` 程式之功能，使之可以轉換出 `cwTEX` 系統格式之字型檔，並且可處理傳統的及 unicode 之字型檔。

上面說明，TrueType 的字型檔案有傳統格式與 unicode 兩種格式。如果欲轉換之 TrueType 字型檔爲傳統格式，請使用 `cwpt1.exe` 轉換，若爲 unicode 格式，請使用 `cwpt1u.exe`。另外，某些公司之字體是以組合方式造字，如華康公司的某些字體，包括 Windows 系統本身的細明體，可能無法作轉換。

`cwpt1` 與 `cwttf` 的主要特色是克服了長久以來 DVIPS 對於 postscript 字形有時無法 partial download 的問題。並且改寫了輸出格式，使得產生的 type1 字形檔案變小；以及加強了一些方便的功能，例如可以任意拉高拉低字形，這對於想在同一種文件中不同廠牌字形的使用者而言，會有極佳的排版效果。其它的功能則是提供了很方便的使用者介面。整套程式開發曠日廢時，在此要特別感謝李果正先生幫忙測試 Linux 版本，你的寶貴意見會在未來努力達成。

20.1.2 轉換方法

取得 TrueType 的中文字型檔之後，首先須確定是 unicode 或傳統規格。大抵而言，若字型檔是 2000 年以前出版的，應該都是傳統規格。TrueType 的中文字型檔是以 `.ttf` 爲延伸檔名。譬如，全真中明體字型檔原名爲 `avmmv.ttf`。請先建立一工作檔案夾，例如 `c:\cwpt1`；將字型檔複製於其內，改名爲 `1.ttf`。字型轉換軟體置於 `c:\texmf\cwtex\util` 檔案夾內，請將 `cwpt1.zip` 亦解壓於 `c:\cwpt1`。

全真中明體爲傳統規格，故應使用 `cwpt1` 程式：

```
c:\cwpt1>cwpt1 n -d 0.94 -n 0.95 -l 0.05 -s 0.167 -e 0.91
```

執行 `cwpt1` 程式時可加入一些選項，以求得最好的結果。爲了解各選項之意義，我們先對字型設計作一點背景說明。我們所轉換之字體是要用於 $\text{T}_{\text{E}}\text{X}$ 系統中，但

每一家字型公司所創造的中文字體之大小其實稍有差異。爲了與 $\text{T}_{\text{E}}\text{X}$ 原有之英數字體與 $\text{cW}_{\text{T}}\text{E}_{\text{X}}$ 中文字體四配，轉換字型檔時，其大小須細心設定。除了調整大小之外，中文 True Type 字體通常設計的較爲方正， $\text{cW}_{\text{T}}\text{E}_{\text{X}}$ 系統所提供的字體則略呈長方形。因此，除了整個字比例性地縮小或放大之外，我們還須將字體調整成長方形。

本例中各選項的意義可簡單說明如下。首先，我們選定轉換後之字體名稱。其次，“-d 0.94”指示將字體等比例縮小94%。下一選項“-n 0.95”指示將字體變成長方形，亦即，高度不變，但寬度變成95%。除此之外，經過測試比較之後，我們發現中文 Windows 字體每一個字的位置比較偏下。故執行轉換程式時，我們加入“-l 0.05”選項，將每一個中文字的位置往上提升原字高度的5%。

第 13.2 節 (頁 259) 介紹中文字體時，曾說明中文字可以作斜體、狹長、與狹長斜體之變化；其中，傾斜角度與狹長之比例可自行設定。以 cwpt1 程式轉換字型檔時，將同時轉換出正體、斜體、狹長、與狹長斜體 4 組字型檔。本例中最後兩個選項“-s 0.167 -e 0.91”即分別選擇斜體字型之傾斜角度，與狹長字體之長寬比例。

由以上的背景資料，以下進一步說明各選項之意義。

- 第 1 選項“n”是新字型名稱。本例之字型取名爲 n，轉換成功之後，若欲使用此字型，指令爲 $\backslash\text{ctxfn}$ 。目前， $\text{cW}_{\text{T}}\text{E}_{\text{X}}$ 系統提供 23 種中文字體，自行轉換之字體最好不要取相同的名字。下一小節將說明可供選用之字體名稱。
- 第 2 選項爲 -d 0.94，其作用是將中文字等比例縮小 94%。此選項之值須設定於中文字之字型定義 (font definition) 檔案內，底下有進一步說明。
- 第 3 選項是 -n 0.95。上一選項將字體等比例縮小，本選項則將方正字體轉換成較爲長方形。其方法是把字體進一步在水平方向縮小 95%。
- 第 4 選項是 -l 0.05，其作用是讓每一個字的位置往上提升原中文字高度的 5%。根據以上之設定，若字體名稱爲 n，字型轉換程式將產生 52 個正體字型檔案：n0.tfm, ..., n51.tfm。
- 第 5-6 選項是 -s 0.167 -e 0.91，設定斜體字之角度與狹長字之水平壓縮比率。本選項設定傾斜角度爲 16.7 度，狹長字之壓縮比率爲 91%。以上之 4 項設定將產生狹長字之字型檔案：ne0.tfm, ..., ne51.tfm，斜體之字型檔案：ns0.tfm, ..., ns51.tfm；以及狹長斜體之字型檔案：nes0.tfm, ...。

程式執行時，視窗內會出現創造字體之訊息。轉換完成之後，`c:\cwpt1` 之下會出現 4 個檔案夾，分別為 `\afm`，`\tfm`，`\type1`，及 `\fd`，每一檔案夾內各有眾多檔案，這些檔案必須分別移至 \TeX 系統之下。以 $\text{MiK}\TeX$ 為例，在 `\texmf\fonts` 檔案夾下，可找到 `\afm`，`\tfm`，及 `\type1` 等 3 個子檔案夾；各檔案夾之下又都有 `\cwtex` 子檔案夾。`cwpt1` 軟體所產生之 3 個子檔案夾內之檔案，必須分別移入對應之 `\cwtex` 子檔案夾內。

最後，請檢視 `c:\cwpt1\fd` 檔案夾內之字型定義檔案，檔案夾內有許多檔案，檔名分別為 `OT1cwn0.fd`，`OT1cwns0.fd`，...。在某些作業系統下，若檔名超過 8 個字，例如 `OT1cwnes15.fd`，檔名可能出現錯亂。若無錯亂情形，請將全部檔案移入下列檔案夾：

```
c:\texmf\tex\latex\cwfond
```

萬一有部分之檔案名字出現錯亂，請直接進入以上之檔案夾內，以文字編輯軟體開啓 `makecwfd.bat`，第一行執行指令為：

```
call cwfont N n 0.94
```

其中，“0.94”參數對應上面執行 `cwpt1.exe` 程式時所加入之“-d 0.94”選項。直接執行 `makecwfd.bat`，即可產生所有之字型定義檔。

以上所有之字型檔案移入各自之檔案夾之後，請更新 \TeX 之檔案資料系統。安裝 $\text{MiK}\TeX$ 者請執行：

```
c:\>initexmf -u
```

安裝 $\text{fp}\TeX$ 者，請執行：

```
c:\>mktexlsr
```

20.1.3 字型對應檔

使用新字型之前，我們還須在中文字型對應檔 `.map` 中加入設定指令。字型設定檔案置於 `c:\texmf\dvips\cwtex` 檔案夾內，以文字編輯軟體開啓 `cwtex.map`，其內有下列定義：


```
m0 CWTEX-M0 <cwtext.enc <m0.pfb
m1 CWTEX-M1 <cwtext.enc <m1.pfb
...
m51 CWTEX-M51 <cwtext.enc <m51.pfb
```

第 1 行指令定義 cwTeX 明體字 $m0$ 之描邊字型來源為 $m0.pfb$ ，其目的是讓預覽與列印程式可以找到所需之字型檔。我們現已轉換出一新字型，名稱為 $\backslash\text{ctxfn}$ ，故須作類似之設定。在 cwtext.map 檔案末端鍵入下列指令：

```
n0 CWTEX-N0 "0.95 ExtendFont" <cwtext.enc <n0.pfb
n1 CWTEX-N1 "0.95 ExtendFont" <cwtext.enc <n1.pfb
...
n51 CWTEX-N51 "0.95 ExtendFont" <cwtext.enc <n51.pfb
```

與明體字不同的是，每一行之設定中加入了 "0.95 ExtendFont" 其功能是指示在預覽與列印時，將中文字縮為長方形。此處之 0.95 對應執行 cwpt1 程式時之 $-n\ 0.95$ 選項。若選項值改變為 0.94， cwtext.map 內之定義也須對應改變。

除了正體字之外，如果我們還要使用狹長字體 $\backslash\text{ctxfne}$ 斜體字 $\backslash\text{ctxfns}$ ，與狹長斜體字 $\backslash\text{ctxfnes}$ 則必須在 cwtext1.map 檔案內加入類似的設定。以第 0 個字型檔字例，須加入之定義為：

```
ns0 CWTEX-N0 "0.95 ExtendFont .167 SlantFont" <cwtext.enc <n0.pfb
ne0 CWTEX-N0 "0.8645 ExtendFont" <cwtext.enc <n0.pfb
nes0 CWTEX-N0 "0.8645 ExtendFont .167 SlantFont" <cwtext.enc <n0.pfb
```

第 1 行指令設定斜體字，角度為 16.7 度。第 2 行指令設定狹長字體，壓縮比率為 0.8645。前面曾經說明，中文 Windows 字體較方正，為了使正體字具長方形特徵，轉換時已作水平壓縮 95%。而狹長字體是進一步水平壓縮 91%，故合計壓縮比率為： $0.91 \times 0.95 = 0.8645$ 。第 3 行指令係合併前兩行指令之設定。

以上的例子說明如何轉換全真中明體，此一字型檔含 13,503 個中文字。某些字型檔則僅含 5,401 個中文字，例如雅坊美工字體。此時，轉換後僅產生 22 個字型檔案。例如，若字體取名為 p ，則 $\backslash\text{type1}$ 檔案夾內僅有 $p0.pfb$ 至 $p21.pfb$ 。此時， .map 檔案內只需要 22 行定義。

以上所轉換的是 Windows 傳統字型檔。若是 unicode 字型檔，其延伸檔名為 .ttc 。例如，華康 unicode 細明體檔名為 Dffn_m3.ttc 。轉換時，先將字型檔複製

表 20.1: cwTeX NI

	cwTeX NI				
b (.)	\ctxfbl	\ctxfb	\ctxfbb	\ctxfbu	\ctfbbx
f (")	\ctxf1	\ctxff			
k ()	\ctxfkl	\ctxfk		\ctxfku	\ctfkx
l ()		\ctxf1	\ctxf1b		
m (p)	\ctxfml	\ctxfm	\ctxfmb	\ctxfmu	\ctfmx
n					
o					
p					
r ()	\ctxfrl	\ctxfr	\ctxfrb	\ctxfriu	\ctfrx

為 1.ttc。若轉換之字體取名為 \ctxfnl，則執行指令例子如下：

```
c:\cwpt1>cwpt1u nl -d 0.94 -n 0.95 -l 0.08 -e 0.91 -s 0.167
```

下一小節即是以轉換出來之字體所排版。

除了 cwpt1 程式之外，我們也可使用 cwttf 程式作轉換。轉換之步驟和以上所述相同，但執行程式時，cwpt1.exe 以 cwttf.exe 替代。若為 unicode 字型檔，則 cwpt1u.exe 以 cwttfu.exe 替代。一般而言，cwpt1 執行時需較長時間，但品質似較佳。

開發測試以上兩套程式，花費甚久時間。但市面上中文字體很多，我們的測試可能仍不完整，歡迎使用者提供改進意見。

20.1.4 選用字體名稱

2dBv, .hΔ [20.1 — cwTeX 5, 5Δ [2J — /5NI, \ctxfm C \ctxfriu, [vff%Δ, ΔffΞ [XU5, Obf.Δ W, , ~JMCEpF5Φ \ctxfnl ; ~u ° Δ

5"1ΨBf, W- "NI — cwTeX Ø5:

```
c:\cwpt1>cwpt1 pb -d 0.94 -n 0.95 -l 0.05 -e 0.91 -s 0.167
```

° v, NI \ctxfpb Δ

20.2 中文字之排序

日常生活中經常使用之中文字大約在三千到四千之間。目前個人電腦系統所提供的每一套中文字型，字數都是 13,053 字。整套字分常用字與次常用字兩部分，前者含 5,401 字。

cwTeX 的基本中文字體 (明體、粗黑體、楷體、圓體與仿宋體) 造滿 13,053 字，但其餘少數字型檔僅含 5,401 字。cwTeX 中文字亦有常用字與非常用字之區分，但編碼順序與 Big-5 不同。cwTeX 常用字有 6,844 個字，分佈於 0–26 個字型檔中，每一字型檔最多可置放 256 個字。以明體字為例，m0 字型檔內之中文字從第 44 排至第 255；m21 字型檔內從 0 排至 238；其餘字型檔都是從 0 排至 255。

cwTeX 系統之次常用字有 6,209 個，放置於字型檔 27–51 之間。其中，第 27–50 字型檔皆存放 256 字，第 51 字型檔的 0–64 存放 65 個次常用字；第 87–255 則存放日文平假名與片假名。若不計入日文字，常用字與次常用字合計為 13,053 字。

如果你的文章中用了某一個字不在上述的範圍之內，執行 cwTeX 時會出現警告訊息。此時，你的文章仍然可以編排，但是在排版的版面上該字將成為空白。缺字可能有兩種情形：第一種是該字可輸入文稿中，但無法預視或列印。第二種情況是連在中文系統下都無法輸入此字，表示此字為 13,503 字以外。

在第一種情況下，我們只要造出中文字，問題即可解決。第二種情況較麻煩，我們還須對所缺之中文字設定中文內碼，並讓中文系統可以鍵入此字。中文 Windows 系統可以造字，請自行參照手冊進行。為了讓 cwTeX 能夠處理新造字，新造字之內碼必須在下列範圍內：FA40-FEFE，8E40-A0FE 及 8140-8DFE；而新造字必須置放於第 52–57 字型檔中。譬如，若所缺字為明體，則一開始之新造字應存於 m52 字型檔內。

若不想在中文系統下也造一對應新字，一個偷懶的解決辦法如下：從中文系統的次常用字中選擇一個幾乎不可能用到的中文字。輸入時以此字替代我們所要造的新字。造好新字之後，將此字之內碼對應到選用替代字上，即可預視或列印正確的文稿。請見下一節之說明。

20.3 造新字

cwTeX 系統之中文字是 Type 1 描邊字型規格。如果有字型設計軟體, 如 **Fontographer**, 即可造 Type 1 描邊字。造好一描邊字型之後, 可由工具程式產生任可尺寸之描點字型檔。使用上較為方便。造字時應注意新造字之大小須與原有字體相當。因此, 理想的方法是以現有字型檔為基礎開始造字。其中細節, 請見各軟體之使用手冊。

造完字之後, 字型軟體可以產生 Type 1 字型檔案, 其中包含各式之字型規格檔案。假設字型檔名為 m52, 我們僅留存 m52.pfb 與 m52.afm 兩檔案, 其餘可刪除。使用 `afm2tfm` 程式由 m52.afm 產生 m52.tfm, 即大功告成。最後, 請將 TeX 字型規格檔 m52.tfm 移入 `\texmf\fonts\fm\cwtex` 檔案夾; Type 1 描邊字型檔 m52.pfb 移入 `\texmf\fonts\type1\cwtex` 檔案夾; Type 1 字型規格檔 m52.afm 則移入 `\texmf\fonts\afm\cwtex` 檔案夾。

20.4 使用新字

欲使用新造字, 我們必須先建立一字元對照檔案 `cwfont.usr`。此一檔案必須置於 `\texmf\cwtex` 檔案夾中。檔案中, 每一行標示一新造字、對應之字型檔名、及字元位置, 其順序為: “新字 字型檔編號 字元位置”。例如:

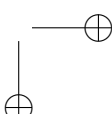
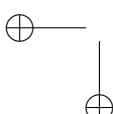
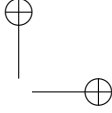
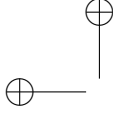
```
涂 52 59 % m
廊 52 1 % m
```

上例中, 「涂」為所造的第一個新字, 置於 m52 字型檔的第 59 個字碼位置; 「廊」則置於第 1 個字碼位置。註銷指令之後為備忘資訊, 提醒自己, 這兩個字僅有明體字型。如果你同時造了明體與粗黑體, 則可標示為 “% m, bb”。

經過以上設定, 轉換中文字時若遇有缺字情形, 如上例中之「涂」字, 轉換程式即檢查 `cwfont.usr` 檔案內是否有字元對應。若發現有對應, cwTeX 即設定取用 m52 字型檔之第 59 個字元排版文稿。

若所缺之字並不在 Big-5 之 13,053 字範圍內, 我們還須在 Windows 中文系統下造對應新字, 才能輸入於文稿中。但是, 中文系統之造字也很麻煩, 因此, cwTeX 提供一偷懶辦法: 從次常用字選用一字作為缺字之對應。再以上的例字說明, 事實上, 「涂」字不在 Big 5 內碼內, 因此無法直接輸入於文稿。假設我們已造好此

中文字，並置於 m52 字型檔第 59 個字碼位置。則 `cwfont.usr` 檔案第 1 行第 1 個字可以用一個確定不會使用之次常用字替代，譬如，「捺」字。輸入文稿時，「涂」之位置即鍵入「捺」；`cwtex` 程式轉換中文時，遇有次常用字會先到 `cwfont.usr` 檔案搜尋是否有字元對應。若有對應，所缺之字即選用 m52 字型檔之第 59 個字排版。如此，雖然文稿所鍵入的不是正確的字，但透過替代字元對應仍可排出正確的結果。



21 錯誤訊息

排版過程不免出現錯誤。錯誤的來源很多：少輸入右大括號，左大括號誤為左圓括號，指令鍵入錯誤，指令的用法錯誤等等。排版時，`cwTeX` 與 `ETEX` 會試著指出錯誤來源。但是，有些錯誤一開始難以判斷其原因及發生處。譬如，以 `\footnote` 排版註解時，若忘了鍵入右大括號，`ETEX` 即設法將其下的文字全部排為註解。最後，因為容納註解文字之空間不足，乃出現 `TeX capacity exceeded ...` 之訊息。而實際上，真正的原因是文稿中少了一個右大括號。

解讀錯誤訊息的能力愈強，我們就能愈快的完成排版工作。底下，21.1 節說明 `cwTeX` 之錯誤與警告訊息，21.2 節說明 `TeX` 與 `ETEX` 的錯誤與警告訊息的意義。21.3 節提示迅速找出錯誤原因的訣竅以供參考。

21.1 `cwTeX` 訊息

`cwTeX` 的錯誤與警告訊息主要是在中文字體方面。執行 `cwTeX` 轉換程式時，若遇上字型檔中沒有中文字，`cwTeX` 會發出警告訊息，說明錯誤發生於檔案的那一行，並將該字留為空白。

以第 3 章之 `test.ctx` 為例，若第 5 行末端之「易」輸入錯誤，變成是字型檔案中沒有的中文字，則轉換中文字時，顯示器上將出現下列警告訊息：

```
! Line 5: X is not character of cfont. Ignore.
```

其中 `X` 為鍵入之中文字。

此外，在轉換中文字時，`cwTeX` 程式會簡單檢查文稿檔案中可能的格式錯誤。首先，輸入指令時左右大括號必須對稱出現。因此，文稿中的右大括號數目若少於左大括號，轉換程式即發出下列訊息：

```
! Too many {'s.
```

反之,若在最後一行(第7行)的 `\end{document}` 指令中,遺漏了左大括號,而使文稿中右大括號數目多於左大括號,則訊息如下:

```
! Line 7 too many }'s.
Press Enter key to continue or x and enter key to quit ? [9)
```

若暫時不想處理此一錯誤,應按 [Enter] 鍵繼續以下的轉換工作。

\TeX 文稿是以 `\end{document}` 結尾,因此, `cw \TeX` 程式會檢查檔案結尾處是否有此指令。輸入時若遺漏左大括號,最後一行變成 `\enddocument}`, `cw \TeX` 會發出下列錯誤訊息:

```
! I can't find \end{document}.
```

\TeX 編排文稿時須使用中文字體規格檔案,附加檔名為 `.tfm`。`cw \TeX` 於轉換中文時,會同時檢查硬碟內是否有所需之中文字體規格檔,若缺乏所需字體規格檔案,即發出底下訊息:

```
! I can't find xx.tfm
```

其中, `xx.tfm` 為所缺之中文字體規格檔案。譬如,若所列檔名為 `b2.tfm`,表示硬碟中並無中黑字體。解決辦法如下:安裝文稿中所使用之字型檔案,或者文稿中不要使用此種字體。

若文稿檔名為 `test.ctx`,以 `cw \TeX` 轉換中文之後,硬碟內將產生 `test.tex` 輔助檔案。但是,若原始文稿之附加檔名取為 `.tex`,例如 `mydoc.tex`,中文轉換後之輔助檔案亦將取為同名之 `mydoc.tex`。遇有此種情況, `cw \TeX` 會將原檔案改為 `mydoc.***`;並發出下列訊息:

```
! Don't use .tex as extension name, please rename your mydoc.tex
```

因此,文稿內若有中文,請勿以 `.tex` 為附加檔名。

除此之外, `cw \TeX` 發出的錯誤訊息主要是提示 `\url` 指令內不得使用中文字。若文稿檔名為 `test.ctx`,中文轉換完畢後, `cw \TeX` 會將執行過程記錄於 `test.xlg` 檔案內。必要時,我們可以開啓此檔以了解細節。

21.2 latex 訊息

執行 latex 時所產生之訊息，部分來自 \LaTeX ，部分來自原始的 \TeX 程式。不過，對一般的使用者而言，此一區分並不重要。重要的是，如何理解警告與錯誤訊息，以更正錯誤。

再以第3章的 `test.ctx` 為例，假設輸入第6行指令時，`\sqrt{\beta}` 誤為 `\sqrt{\beta$`，亦即少鍵入右大括號。在轉換中文時，雖然 `cxTeX` 會發出如上一節所述之警告訊息，但仍然會將全文轉換完畢。但接下來，`latex` 編排時，顯示器上將出現下列的錯誤訊息：

```
! Missing } inserted.
<inserted text>
      }
1.6 ...S\cH23}, {\MaS\cH224} $\sqrt{\beta$
                                     , {\MeS\cH171}\z...
?
```

第1行！符號之後的文字表示錯誤的原因，或 \LaTeX 所自動採取的補救方法。此例中訊息內容為 `Missing } inserted.`，表示 \LaTeX 肯定錯誤的原因是少鍵入一個右大括號，因此自行將漏掉的大括號插入。

第4行開頭的訊息 `1.6` 說明指令錯誤發生於第6行 (line 6)。該行末端之 `... \beta$` 與下一行開頭處之 `, {\MeS...` 指明錯誤發生所在。拿以上兩行與原始 `test.ctx` 檔案之第6行比較，我們發現原檔案之第6行的中文字在錯誤訊息中變成奇怪的符號，如 `{\MaS\cH224}`，`{\MeS\cH171}` 等等。事實上，這些符號正是原來中文字經過轉換之後的格式。

錯誤訊息的最後一行之 `?` 號表示 `latex` 正等著使用者之動作。此時，如果我們鍵入另一個 `?` 號，顯示器上即出現下列訊息：

```
Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
?
```

若鍵入 `[Enter]`，`latex` 將繼續執行，直到遇到下一個錯誤才停止。如果不再碰到

錯誤,即排版至最後一頁。

若鍵入下列各鍵,其效果如下:

- S latex 將繼續編排文稿至最後一頁,中間即使再有其他錯誤也不停止。所有之排版訊息會存於 test.log 檔案中,稍後可以文字輸入軟體閱讀之。
- R 與 S 選項類似。但 S 選項若遇有找不到檔案之情況仍會停下來等待進一步之指示;此選項則不停。
- Q 與按 R 鍵類似。但前者會使排版訊息出現於顯示器上,此一選項則使訊息只存於 test.log 中,不出現在顯示器上。
- H 求助選項,latex 將試著進一步分析錯誤之原因,並提出解決之道。
- X 要求立即停止編排,並退出排版程式。

排版時,latex 所產生的訊息中,最重要的是錯誤發生於那幾行。若能確認錯誤之所在,通常我們很快就能找出問題原因。不過,如本章開頭的 \footnote 例子所示,有時候,latex 所指示的行數不一定是錯誤的源頭,可能只是排版程式無法繼續執行之處。大部分的錯誤訊息內容都很清楚,以下只舉幾個常見的訊息略加說明。

輸入 test.ctx 時,若第 6 行之 $\sqrt{\beta}$ 誤為 $\sqrt{\beta}$,將出現下列錯誤訊息:

```
! Undefined control sequence.
<recently read> \sqrt

1.6  $\sqrt{\beta}$ .
?
```

latex 不會說鍵入的指令錯誤,只會提醒說 $\sqrt{\beta}$ 指令未曾無定義過。另外一個類似的錯誤訊息為:

```
! LaTeX Error: Environment ... undefined.
```

此訊息指示文稿中所使用之指令環境並無定義;最可能的原因仍然是指令環境之名字輸入錯誤。

```
! LaTeX Error: Can be used only in preamble.
```

文稿內以 `\usepackage` 指令引用巨集套件時，應下於全文設定區 (preamble)，亦即在 `\begin{document}` 指令之前。若指令誤下於 `\begin{document}` 指令之後，排版軟體即發出上列訊息。

```
! TeX capacity exceeded, sorry ...
```

依字面解釋，此項訊息說明 \TeX 程式的空間不足，無法處理複雜的文稿。但是，如本章一開頭的例子所示，真正原因通常是忘了在 `\footnote{...}` 指令中加上右大括號。

另一個易於出現上述錯誤訊息的情況如下。若文稿內有許多的圖表，且每一圖表都置於浮動圖表 (`figure` 或 `table`) 指令環境內，也容易產生此項錯誤訊息。事實上，如果大部分圖表占一整頁空間，我們可直接以指令控制圖表出現之位置，不須借助於浮動圖表指令環境。另外，一般的 \TeX 版本只允許在一文稿中使用 256 種字體。如果中文文稿內使用太多的字型檔，也可能出現此項錯誤訊息。解決辦法之一是將文稿拆成幾個部分，分開編排；或者使用 `em \TeX` 系統之 `htex386` 程式排版。

```
! I can't find file '...'
Please type another input file name =
```

執行 `latex` 時須鍵入文稿檔名；或者文稿內所引用之巨集檔案可能集中於另一檔案內。若文稿檔名鍵入錯誤，或者置於某檔案夾中但 `latex` 無法自動找到，此時即產生以上之訊息。解決的辦法是輸入正確檔名。萬一硬碟中無任何 `.tex` 檔可以輸入，可試著輸入 `null`，此為一虛擬檔案，`latex` 應會繼續排版動作。

```
! Fatal format error; I'm stymied
```

此一訊息通常出現於新裝設 `cw \TeX` 系統或更新版本之後。`latex` 執行時須從硬碟中讀取某些格式檔案 (format file)。若格式檔案的版本不對，或者設定之檔案夾錯誤，即出現上述訊息。請再確認安裝是否正確。

21.3 確認錯誤來源

偵測錯誤的第一步是確認錯誤之所在。 \TeX 文稿是以 `\end{document}` 結尾，文稿任何地方出現此一訊息，其後的文字或指令都不再處理。因為 `latex` 所提供

的錯誤訊息都會指出錯誤發生的那一行，因此我們可以在該行之後兩三行處下 `\end{document}` 指令，重新排版，看看錯誤訊息是否仍與原來的相同。訊息若相同，表示錯誤是發生於該日之前。

接著，我們由此行往上尋找錯誤。若來回數遍仍無所獲，可以試著在此行之前十數行處再下 `\end{document}` 指令。重新排版之後，若錯誤訊息已消失，表示錯誤是發行於兩個 `\end{document}` 指令當中。以此方法，逐步縮小範圍，即不難確認錯誤所在。

22 資源

本書所介紹的軟體與巨集套件大部分都仍繼續發展中，隨時可能更新。此外，你的排版工作可能需要其他的工具軟體或巨集套件。網路上有豐富的 T_EX 軟體資源，以下分門別類，稍加說明。

- cwT_EX 網站

cwT_EX 設有一網站，存放與光碟內容相同或更新的版本，網址為：

`http://ceiba.cc.ntu.edu.tw/tmwu`

除了存放更新軟體之外，cwT_EX 網站另闢有討論區，供使用者交換心得。由此網站可連結到其他相關網站，例如，中文 T_EX 之 BBS 討論站：

`telnet: 140.112.18.32`

- T_EX Users Group 與 CTAN

國際間之 T_EX 使用者成立 T_EX Users Group，簡稱為 TUG，網址為：

`http://www.tug.org`

這是 T_EX 的官方網站。收集 T_EX 軟體最完整的網站是 CTAN，全名為 Comprehensive T_EX Archive Network。國內複製了 CTAN 全部或部分資料的網站包括：

`ftp://ftp.ccu.edu.tw/pub1/tex`
`ftp://ftp1.sinica.edu.tw/pub2/tex`

- T_EX 討論區與常見問題解答

使用 T_EX 或 L^AT_EX 碰到問題時，你可以到底下的討論區提問，

`http://groups.google.com/groups?hl=zh-Tw&group=comp.text.tex`

專家會迅速提供解答。

英國的 T_EX Users Group 特別整理常見問題之解答, 存於下列網站上:

<http://www.tex.ac.uk/cgi-bin/texfaq2html>

碰到問題時, 可以試由此尋找答案。

- 說明檔

L^AT_EX 所有的巨集套件都有其作者所寫的使用說明, 在 MiK_TE_X 系統中, 使用說明檔收錄於 `c:\texmf\doc` 檔案夾內。其中, 一般觀念性的介紹則置於:

`c:\texmf\doc\guides`

- MiK_TE_X 與 fp_TE_X 網站

MiK_TE_X 官方網站為:

<http://www.miktex.org>

如果你原先選擇較小安裝, 現想要加裝或更新 T_EX 巨集套件, 可直接透過 MiK_TE_X 系統為之。請由 Windows 之

開始 | 程式集 | MiK_TE_X | MiK_TE_X Options | Package | Other

點選欲加裝之套件。如果是要更新 MiK_TE_X 系統軟體, 請由

開始 | 程式集 | MiK_TE_X | MiK_TE_X Update Wizard

fp_TE_X 官方網站為:

<http://www.fptex.org>

參考書目

- Bringinghurst, Robert (1996), *The Elements of Typographic Style*, Vancouver: Hartley & Marks, 2 edn.
 [字體設計家 Hermann Zapf 認為「所有桌上排版者都應閱讀此書」。雖然本書內容是以英文排版為主。不過, 觸類旁通, 書內所討論的原則對於中文排版仍有相當價值。] [5.3.2](#), [6.4.1](#), [13.1.2](#), [18](#)
- Goossens, Michel, Frank Mittelbach, and Alexander Samarin (1994), *The L^AT_EX Companion*, Reading, Massachusetts: Addison Wesley.
 [L^AT_EX2e 工作小組所寫的參考手冊, 介紹許多實用的巨集套件以及新版 L^AT_EX 之功能。] [5](#), [5.4](#), [10.4.1](#), [12.2](#), [14.1](#), [17.4.3](#), [18.3.1](#)
- (1997), “Higher Mathematics”,
 [*The L^AT_EX Companion*, 第 8 章之更新版本, 收錄於
`c:\texmf\doc\guides\companion`。] [9](#)
- Goossens, Michel and Sebastian Rahtz (1999), *L^AT_EX Web Companion*, Reading, Massachusetts: Addison Wesley.
 [介紹如何將 L^AT_EX 文件轉換為網路文件。] [19](#)
- Goossens, Michel, Sebastian Rahtz, and Frank Mittelbach (1997), *L^AT_EX Graphics Companion*, Reading, Massachusetts: Addison Wesley.
 [L^AT_EX2e 工作小組所寫之參考手冊, 主要介紹圖形巨集套件, 特別是如何使用 PostScript 語言。] [9.1](#), [9.3.5](#), [10.6.4](#), [12.1.1](#), [12.2](#), [12.3](#), [12.7](#)
- Grätzer, George (2000), *Math into L^AT_EX*, Boston: Birkhäuser.
 [主要介紹 A_MS-L^AT_EX 巨集套件。] [9](#)
- Knuth, Donald E. (1990), *The T_EX Book*, Reading, Massachusetts: Addison Wesley.
 [T_EX 的起源, 進階使用者之聖經。] [14](#)
- (1999), *Digital Typography*, Leland, California: Stanford Junior University.
 [Knuth 細數 T_EX 與 METAFONT 系統發展過程的點點滴滴, 生動有趣。] [1](#)

- Kopka, Helmut and Patrick W. Daly (1995), *A Guide to L^AT_EX2_ε*, Reading, Massachusetts: Addison Wesley, 2 edn.
[詳盡介紹 L^AT_EX 指令, 可以替代 Lamport (1994) 之使用手冊。] [5](#), [8.4](#), [14](#), [14.1](#), [15.1.2](#)
- Lamport, Leslie (1994), *L^AT_EX: A Document Preparation System*, Reading, Massachusetts: Addison Wesley, 2 edn.
[L^AT_EX 巨集套件原創者所寫之使用手冊。] [5](#)
- Reckdahl, Keith (1997), “Using Imported Graphics in L^AT_EX2_ε”,
[詳細說明如何引用 Postscript 圖形於 L^AT_EX 文稿內。收錄於
`c:\texmf\doc\guides\epslatex` 。] [10.4.1](#), [10.4.2](#), [11.1](#)
- Taylor, Philip (1994–1995), “Book Design for T_EX Users”, *Gust*,
[說明排版書籍的基本原則, 精簡扼要。分兩期載於 *GUST* 雜誌, 收錄於
`c:\texmf\cwtex\help` 。] [6.4.1](#)
- Wilson, Peter (2001), “The Memoir Class for Configurable Typesetting User Guide”,
[memoir 巨集套件之說明檔, 收錄於 `c:\texmf\tex\latex\memoir` 。]
[18](#)

數學符號索引

排版指令索引請見 399 頁。

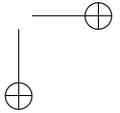
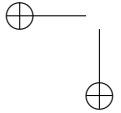
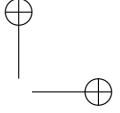
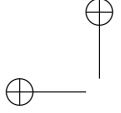
<code>\acute</code> (\acute{o}), 168	<code>\Big</code> , 177	<code>\bowtie</code> (\bowtie), 163
<code>\aleph</code> (\aleph), 167	<code>\bigcap</code> (\bigcap), 161	<code>\Box</code> (\square), 167, 218
<code>\alpha</code> (α), 31, 158	<code>\bigcirc</code> (\bigcirc), 166	<code>\boxdot</code> (\boxdot), 166
<code>\amalg</code> (\amalg), 166	<code>\bigcup</code> (\bigcup), 161	<code>\boxminus</code> (\boxminus), 166
<code>\angle</code> (\angle), 167	<code>\bigg</code> , 177	<code>\boxplus</code> (\boxplus), 166
<code>\approx</code> (\approx), 163	<code>\Bigg</code> , 177	<code>\boxtimes</code> (\boxtimes), 166
<code>\approxeq</code> (\approx), 164	<code>\bigodot</code> (\bigodot), 161	<code>\breve</code> (\breve{o}), 168
<code>\arccos</code> , 160	<code>\bigoplus</code> (\bigoplus), 161	<code>\bullet</code> (\bullet), 166
<code>\arcsin</code> , 160	<code>\bigotimes</code> (\bigotimes), 161	<code>\bumpeq</code> (\bumpeq), 164
<code>\arctan</code> , 160	<code>\bigsqcup</code> (\bigsqcup), 161	<code>\Bumpeq</code> (\Bumpeq), 164
<code>\arg</code> , 160	<code>\bigstar</code> (\bigstar), 167	<code>\cap</code> (\cap), 166
<code>\ast</code> ($*$), 166	<code>\bigtriangledown</code> (\bigtriangledown), 166	<code>\Cap</code> (\cap), 166
<code>\asym</code> (\asymp), 163	<code>\bigtriangleup</code> (\bigtriangleup), 166	<code>\cdot</code> (\cdot), 166
<code>\backepsilon</code> (ϵ), 164	<code>\biguplus</code> (\biguplus), 161	<code>\cdots</code> (\cdots), 170, 171
<code>\backprime</code> (\backprime), 167	<code>\bigvee</code> (\bigvee), 161	<code>\centerdot</code> (\cdot), 166
<code>\backsim</code> (\backsim), 164	<code>\bigwedge</code> (\bigwedge), 161	<code>\check</code> (\check{o}), 168
<code>\backsimeq</code> (\backsimeq), 164	<code>\blacklozenge</code> (\blacklozenge), 167	<code>\chi</code> (χ), 158
<code>\backslash</code> (\backslash), 167, 177	<code>\blacksquare</code> (\blacksquare), 167	<code>\circ</code> (\circ), 166
<code>\bar</code> (\bar{o}), 168	<code>\blacktriangle</code> (\blacktriangle), 167	<code>\circeq</code> ($\overset{=}{\circ}$), 164
<code>\barwedge</code> ($\bar{\wedge}$), 166	<code>\blacktriangledown</code> (\blacktriangledown), 167	<code>\circlearrowleft</code> (\circlearrowleft), 162
<code>\Bbbk</code> (\mathbb{k}), 167	<code>\blacktriangleright</code> (\blacktriangleright), 164	<code>\circlearrowright</code> (\circlearrowright), 162
<code>\because</code> (\because), 164	<code>\blacktriangleleft</code> (\blacktriangleleft), 164	<code>\circledast</code> (\circledast), 166
<code>\beta</code> (β), 40, 158	<code>\blacktriangleright</code> (\blacktriangleright), 164	<code>\circledcirc</code> (\circledcirc), 166
<code>\beth</code> (\beth), 167	<code>\blacktriangleright</code> (\blacktriangleright), 164	<code>\circleddash</code> (\circleddash), 166
<code>\between</code> (\between), 164	<code>\bot</code> (\perp), 167	<code>\circledS</code> (\circledS), 167
<code>\big</code> , 177		<code>\clubsuit</code> (\clubsuit), 167

<code>\complement</code> (\complement), 167	<code>\dot</code> ($\dot{}$), 168	<code>\ge</code> (\geq), 163
<code>\cong</code> (\cong), 163	<code>\doteq</code> (\doteq), 163	<code>\geq</code> (\geq), 163
<code>\coprod</code> (\coprod), 161	<code>\Doteq</code> (\doteq), 164	<code>\geqq</code> (\geqq), 164
<code>\cos</code> , 160	<code>\doteqdot</code> (\doteqdot), 164	<code>\geqslant</code> (\geqslant), 164
<code>\cosh</code> , 160	<code>\dotplus</code> ($\dot{+}$), 166	<code>\gets</code> (\leftarrow), 161
<code>\cot</code> , 160	<code>\doublebarwedge</code> ($\overline{\wedge}$), 166	<code>\gg</code> (\gg), 163
<code>\coth</code> , 160	<code>\doublecap</code> ($\overline{\cap}$), 166	<code>\ggg</code> (\ggg), 164
<code>\csc</code> , 160	<code>\doublecup</code> ($\overline{\cup}$), 166	<code>\gggtr</code> (\gggtr), 164
<code>\cup</code> (\cup), 166	<code>\downarrow</code> (\downarrow), 162, 177	<code>\gimel</code> (\beth), 167
<code>\Cup</code> (\cup), 166	<code>\Downarrow</code> (\Downarrow), 162, 177	<code>\gnapprox</code> (\gtrsim), 165
<code>\curlyeqprec</code> (\curlyeqprec), 164	<code>\downdownarrows</code> (\Downarrow), 162	<code>\gneq</code> (\gtrsim), 165
<code>\curlyeqsucc</code> (\curlyeqsucc), 164	<code>\downharpoonleft</code> (\lrcorner), 162	<code>\gneqq</code> (\gtrneqq), 165
<code>\curlyvee</code> (\curlyvee), 166	<code>\downharpoonright</code> (\llcorner), 162	<code>\gnsim</code> (\gtrsim), 165
<code>\curlywedge</code> (\curlywedge), 166	<code>\ell</code> (ℓ), 167	<code>\grave</code> ($\grave{}$), 168
<code>\curvearrowleft</code> (\curvearrowleft), 162	<code>\emptyset</code> (\emptyset), 167	<code>\gtrapprox</code> (\gtrsim), 164
<code>\curvearrowright</code> (\curvearrowright), 162	<code>\epsilon</code> (ϵ), 158	<code>\gtrdot</code> (\gtrdot), 164
<code>\dagger</code> (\dagger), 166	<code>\eqcirc</code> (\equiv), 164	<code>\gtreqless</code> (\gtrless), 164
<code>\daleth</code> (\beth), 167	<code>\eqsim</code> (\simeq), 164	<code>\gtreqqless</code> (\gtrless), 164
<code>\dashv</code> (\dashv), 163	<code>\eqslantgtr</code> (\gtrsim), 164	<code>\gtrless</code> (\gtrless), 164
<code>\ddagger</code> (\ddagger), 166	<code>\eqslantless</code> (\lesssim), 164	<code>\gtrsim</code> (\gtrsim), 164
<code>\ddot</code> ($\ddot{}$), 168	<code>\equiv</code> (\equiv), 163	<code>\gvertneqq</code> (\gtrneqq), 165
<code>\ddots</code> (\ddots), 171	<code>\eta</code> (η), 158	<code>\hat</code> ($\hat{}$), 168
<code>\deg</code> , 160	<code>\eth</code> (\eth), 167	<code>\hbar</code> (\hbar), 167
<code>\delta</code> (δ), 31, 158	<code>\exists</code> (\exists), 167	<code>\heartsuit</code> (\heartsuit), 167
<code>\Delta</code> (Δ), 158	<code>\exp</code> , 160	<code>\hom</code> , 160
<code>\det</code> , 160	<code>\fallingdotseq</code> (\fallingdotseq), 164	<code>\hookleftarrow</code> (\hookleftarrow), 162
<code>\diagdown</code> (\diagdown), 167	<code>\Finv</code> (\Finv), 167	<code>\hookrightarrow</code> (\hookrightarrow), 162
<code>\diagup</code> (\diagup), 167	<code>\flat</code> (\flat), 167	<code>\hslash</code> (\hbar), 167
<code>\diamond</code> (\diamond), 166	<code>\forall</code> (\forall), 167	<code>\Im</code> (\Im), 167
<code>\Diamond</code> (\Diamond), 167	<code>\frac</code> , 156	<code>\imath</code> (\imath), 167, 168
<code>\diamondsuit</code> (\diamondsuit), 167	<code>\frown</code> (\frown), 163	<code>\in</code> (\in), 163
<code>\digamma</code> (\digamma), 167	<code>\Game</code> (\Game), 167	<code>\inf</code> , 160
<code>\dim</code> , 160	<code>\gamma</code> (γ), 158	<code>\infty</code> (∞), 167
<code>\div</code> (\div), 166	<code>\Gamma</code> (Γ), 158	<code>\int</code> (\int), 160, 161
<code>\divideontimes</code> (\div), 166	<code>\gcd</code> , 160	<code>\intercal</code> (\intercal), 166

<code>\iota</code> (ι), 158	<code>\lesssim</code> (\lesssim), 164	<code>\max</code> , 160
<code>\jmath</code> (j), 167, 168	<code>\lfloor</code> (\lfloor), 177	<code>\measuredangle</code> (\sphericalangle), 167
<code>\Join</code> (\bowtie), 163	<code>\lg</code> , 160	<code>\mho</code> (\O), 167
<code>\kappa</code> (κ), 158	<code>\lhd</code> (\triangleleft), 166	<code>\mid</code> ($ $), 163
<code>\ker</code> , 160	<code>\lim</code> , 160	<code>\min</code> , 160
<code>\lambda</code> (λ), 158	<code>\liminf</code> , 160	<code>\models</code> (\models), 163
<code>\Lambda</code> (Λ), 158	<code>\limsup</code> , 160	<code>\mp</code> (\mp), 166
<code>\langle</code> (\langle), 177	<code>\ll</code> (\ll), 163	<code>\mu</code> (μ), 158
<code>\lceil</code> (\lceil), 177	<code>\llcorner</code> (\llcorner), 167	<code>\multimap</code> (\multimap), 162
<code>\ldots</code> (\dots), 170	<code>\Lleftarrow</code> (\Lleftarrow), 162	<code>\nabla</code> (∇), 167
<code>\le</code> (\leq), 162	<code>\lll</code> (\lll), 164	<code>\natural</code> (\natural), 167
<code>\leadsto</code> (\leadsto), 162	<code>\llless</code> (\llless), 164	<code>\ncong</code> ($\not\cong$), 165
<code>\left.</code> , 178	<code>\ln</code> , 160	<code>\nearrow</code> (\nearrow), 162
<code>\leftarrow</code> (\leftarrow), 162	<code>\lnapprox</code> (\lnapprox), 165	<code>\neg</code> (\neg), 167
<code>\Leftarrow</code> (\Leftarrow), 162	<code>\lneq</code> (\lneq), 165	<code>\neq</code> (\neq), 163
<code>\leftarrowtail</code> (\leftarrowtail), 162	<code>\lneqq</code> (\lneqq), 165	<code>\nexists</code> (\nexists), 167
<code>\leftharpoondown</code> (\leftharpoondown), 162	<code>\lnsim</code> (\lnsim), 165	<code>\ngeq</code> (\ngeq), 165
<code>\leftharpoonup</code> (\leftharpoonup), 162	<code>\log</code> , 160	<code>\ngeqq</code> (\ngeqq), 165
<code>\leftleftarrows</code> (\leftleftarrows), 162	<code>\longleftarrow</code> (\longleftarrow), 162	<code>\ngeqslant</code> (\ngeqslant), 165
<code>\leftrightarrows</code> (\leftrightarrows), 162	<code>\Longleftarrow</code> (\Longleftarrow), 162	<code>\ngtr</code> (\ngtr), 165
<code>\Leftrightarrow</code> (\Leftrightarrow), 162	<code>\longlefrightharpoonrightarrow</code> (\longleftrightarrow), 162	<code>\ni</code> (\ni), 163
<code>\leftrightharpoons</code> (\leftrightharpoons), 162	<code>\Longlefrightharpoonrightarrow</code> (\longleftrightarrow), 162	<code>\nleftarrow</code> (\nleftarrow), 162
<code>\leftrightsquigarrow</code> (\leftrightsquigarrow), 162	<code>\longmapsto</code> (\longmapsto), 162	<code>\nLeftarrow</code> (\nLeftarrow), 162
<code>\leftthreetimes</code> (\leftthreetimes), 166	<code>\longrightarrow</code> (\longrightarrow), 162	<code>\nlefrightharpoonrightarrow</code> ($\nlefrightharpoonrightarrow$), 162
<code>\leq</code> (\leq), 163	<code>\Longrightarrow</code> (\Longrightarrow), 162	<code>\nleq</code> (\nleq), 165
<code>\leqq</code> (\leqq), 164	<code>\looparrowleft</code> (\looparrowleft), 162	<code>\nleqq</code> (\nleqq), 165
<code>\leqslant</code> (\leqslant), 164	<code>\looparrowright</code> (\looparrowright), 162	<code>\nleqslant</code> (\nleqslant), 165
<code>\lessapprox</code> (\lessapprox), 164	<code>\lozenge</code> (\lozenge), 167	<code>\nless</code> (\nless), 165
<code>\lessdot</code> (\lessdot), 164	<code>\lrcorner</code> (\lrcorner), 167	<code>\nmid</code> (\nmid), 165
<code>\lesseqgtr</code> (\lesseqgtr), 164	<code>\Lsh</code> (\Lsh), 162	<code>\notin</code> (\notin), 163
<code>\lesseqgtr</code> (\lesseqgtr), 164	<code>\ltimes</code> (\ltimes), 166	<code>\notin</code> (\notin), 163
<code>\lessgtr</code> (\lessgtr), 164	<code>\lvertneqq</code> (\lvertneqq), 165	<code>\nparallel</code> (\nparallel), 165
	<code>\mapsto</code> (\mapsto), 162	<code>\nprec</code> (\nprec), 165
		<code>\npreceq</code> (\npreceq), 165

<code>\nrightarrow</code> (\nrightarrow), 162	<code>\partial</code> (∂), 167	<code>\rightleftarrows</code> (\rightleftarrows), 162
<code>\nRrightarrow</code> (\nRrightarrow), 162	<code>\perp</code> (\perp), 163	<code>\rightleftharpoons</code> (\rightleftharpoons), 162
<code>\nshortmid</code> (\nshortmid), 165	<code>\phi</code> (ϕ), 158	<code>\rightrightarrows</code> (\rightrightarrows), 162
<code>\nshortparallel</code> (\nshortparallel), 165	<code>\Phi</code> (Φ), 158	<code>\rightsquigarrow</code> (\rightsquigarrow), 162
<code>\nsim</code> (\nsim), 165	<code>\pi</code> (π), 158	<code>\rightthreetimes</code> (\rightthreetimes), 166
<code>\nsubseteq</code> (\nsubseteq), 165	<code>\Pi</code> (Π), 158	<code>\risingdotseq</code> (\risingdotseq), 164
<code>\nsubseteqq</code> (\nsubseteqq), 165	<code>\pitchfork</code> (\pitchfork), 164	<code>\Rrightarrow</code> (\Rrightarrow), 162
<code>\nsucc</code> (\nsucc), 165	<code>\pm</code> (\pm), 166	<code>\Rsh</code> (\Rsh), 162
<code>\nsucceq</code> (\nsucceq), 165	<code>\Pr</code> , 160	<code>\rtimes</code> (\rtimes), 166
<code>\nsupseteq</code> (\nsupseteq), 165	<code>\prec</code> (\prec), 163	<code>\searrow</code> (\searrow), 162
<code>\nsupseteqq</code> (\nsupseteqq), 165	<code>\preccurlyeq</code> (\preccurlyeq), 164	<code>\sec</code> , 160
<code>\ntriangleleft</code> (\ntriangleleft), 165	<code>\preceq</code> (\preceq), 163	<code>\setminus</code> (\setminus), 166
<code>\ntrianglelefteq</code> (\ntrianglelefteq), 165	<code>\precnapprox</code> (\precnapprox), 164	<code>\sharp</code> (\sharp), 167
<code>\ntriangleright</code> (\ntriangleright), 165	<code>\precneq</code> (\precneq), 165	<code>\shortmid</code> (\shortmid), 164
<code>\ntrianglerighteq</code> (\ntrianglerighteq), 165	<code>\precnsim</code> (\precnsim), 165	<code>\shortparallel</code> (\shortparallel), 164
<code>\nu</code> (ν), 158	<code>\precsim</code> (\precsim), 164	<code>\sigma</code> (σ), 158
<code>\nvDash</code> (\nvDash), 165	<code>\prime</code> (\prime), 167	<code>\Sigma</code> (Σ), 158
<code>\nVDash</code> (\nVDash), 165	<code>\prod</code> (\prod), 161	<code>\sim</code> (\sim), 163
<code>\nVDash</code> (\nVDash), 165	<code>\propto</code> (\propto), 163	<code>\simeq</code> (\simeq), 163
<code>\nwarrow</code> (\nwarrow), 162	<code>\psi</code> (ψ), 158	<code>\sin</code> , 160
<code>o</code> (o), 158	<code>\Psi</code> (Ψ), 158	<code>\sinh</code> , 160
<code>\odot</code> (\odot), 166	<code>\rangle</code> (\rangle), 177	<code>\smallfrown</code> (\smallfrown), 164
<code>\oint</code> (\oint), 161	<code>\rceil</code> (\rceil), 177	<code>\smallsetminus</code> (\smallsetminus), 166
<code>\omega</code> (ω), 158	<code>\Re</code> (\Re), 167	<code>\smallsmile</code> (\smallsmile), 164
<code>\Omega</code> (Ω), 158	<code>\restriction</code> (\restriction), 162	<code>\smile</code> (\smile), 163
<code>\ominus</code> (\ominus), 166	<code>\rfloor</code> (\rfloor), 177	<code>\spadesuit</code> (\spadesuit), 167
<code>\oplus</code> (\oplus), 166	<code>\rhd</code> (\rhd), 166	<code>\sphericalangle</code> (\sphericalangle), 167
<code>\oslash</code> (\oslash), 166	<code>\rho</code> (ρ), 158	<code>\sqcap</code> (\sqcap), 166
<code>\otimes</code> (\otimes), 166	<code>\right.</code> , 178	<code>\sqcup</code> (\sqcup), 166
<code>\overbrace</code> ($\overbrace{a+b}$), 168	<code>\rightarrow</code> (\rightarrow), 162	<code>\sqsubset</code> (\sqsubset), 163, 164
<code>\overleftarrow</code> , 161	<code>\Rrightarrow</code> (\Rrightarrow), 162	<code>\sqsupset</code> (\sqsupset), 163, 164
<code>\overline</code> (\overline{a}), 168	<code>\rightarrowtail</code> (\rightarrowtail), 162	<code>\sqsupseteq</code> (\sqsupseteq), 163
<code>\overrightarrow</code> , 161	<code>\rightharpoondown</code> (\rightharpoondown), 162	<code>\square</code> (\square), 167
<code>\parallel</code> (\parallel), 163	<code>\rightharpoonup</code> (\rightharpoonup), 162	<code>\star</code> (\star), 166

<code>\subset</code> (\subset), 163	<code>\thicksim</code> (\sim), 164	<code>\varkappa</code> (\varkappa), 167
<code>\Subset</code> (\Subset), 164	<code>\tilde</code> (\tilde), 168	<code>\varnothing</code> (\emptyset), 167
<code>\subseteq</code> (\subseteq), 163	<code>\times</code> (\times), 166	<code>\varphi</code> (φ), 158
<code>\subseteqeq</code> (\subseteq), 164	<code>\to</code> (\rightarrow), 162	<code>\varpi</code> (ϖ), 158
<code>\subsetneq</code> (\subsetneq), 165	<code>\top</code> (\top), 167	<code>\varpropto</code> (\propto), 164
<code>\subsetneqq</code> (\subsetneqq), 165	<code>\triangle</code> (\triangle), 167	<code>\varrho</code> (ϱ), 158
<code>\succ</code> (\succ), 163	<code>\triangledown</code> (∇), 167	<code>\varsigma</code> (ς), 158
<code>\succapprox</code> (\succapprox), 164	<code>\triangleleft</code> (\triangleleft), 166	<code>\varsubsetneq</code> (\varsubsetneq), 165
<code>\succcurlyeq</code> (\succcurlyeq), 164	<code>\trianglelefteq</code> (\trianglelefteq), 164	<code>\varsubsetneqq</code> (\varsubsetneqq), 165
<code>\succeq</code> (\succeq), 163	<code>\trianglelefteq</code> (\trianglelefteq), 164	<code>\varsupsetneq</code> (\varsupsetneq), 165
<code>\succnapprox</code> (\succnapprox), 165	<code>\triangleright</code> (\triangleright), 166	<code>\varsupsetneqq</code> (\varsupsetneqq), 165
<code>\succneqq</code> (\succneqq), 165	<code>\trianglerighteq</code> (\trianglerighteq), 164	<code>\vartheta</code> (ϑ), 158
<code>\succnsim</code> (\succnsim), 165	<code>\twoheadleftarrow</code> (\twoheadleftarrow), 162	<code>\vartriangle</code> (\triangle), 167
<code>\succsim</code> (\succsim), 164	<code>\twoheadrightarrow</code> (\twoheadrightarrow), 162	<code>\vartriangleleft</code> (\triangleleft), 164
<code>\sum</code> (\sum), 160, 161		<code>\vartriangleright</code> (\triangleright), 164
<code>\sup</code> , 160	<code>\ulcorner</code> (\ulcorner), 167	<code>\vdash</code> (\vdash), 163
<code>\supset</code> (\supset), 163	<code>\underbrace</code> ($\underbrace{a+b}$), 168	<code>\VDash</code> (\Vdash), 164
<code>\Supset</code> (\Supset), 164	<code>\unlhd</code> (\triangleleft), 166	<code>\Vdash</code> (\Vdash), 164
<code>\supseteq</code> (\supseteq), 163	<code>\unrhd</code> (\triangleright), 166	<code>\vdots</code> (\vdots), 171
<code>\supseteqeq</code> (\supseteq), 164	<code>\uparrow</code> (\uparrow), 162, 177	<code>\vec</code> (\vec), 168
<code>\supsetneq</code> (\supsetneq), 165	<code>\Uparrow</code> (\Uparrow), 162, 177	<code>\vee</code> (\vee), 166
<code>\supsetneqq</code> (\supsetneqq), 165	<code>\updownarrow</code> (\updownarrow), 162, 177	<code>\veebar</code> (\veebar), 166
<code>\surd</code> (\surd), 167	<code>\Updownarrow</code> (\Updownarrow), 162, 177	<code>\Vvdash</code> (\Vvdash), 164
<code>\swarrow</code> (\swarrow), 162	<code>\upharpoonleft</code> (\upharpoonleft), 162	<code>\wedge</code> (\wedge), 166
<code>\tan</code> , 160	<code>\upharpoonright</code> (\upharpoonright), 162	<code>\widehat</code> (\widehat), 168
<code>\tanh</code> , 160	<code>\uplus</code> (\uplus), 166	<code>\widetilde</code> (\widetilde), 168
<code>\tau</code> (τ), 158	<code>\upsilon</code> (υ), 158	<code>\wp</code> (\wp), 167
<code>\therefore</code> (\therefore), 164	<code>\Upsilon</code> (Υ), 158	<code>\wr</code> (\wr), 166
<code>\theta</code> (θ), 158	<code>\upuparrows</code> (\upuparrows), 162	<code>\xi</code> (ξ), 158
<code>\Theta</code> (Θ), 158	<code>\urcorner</code> (\urcorner), 167	<code>\Xi</code> (Ξ), 158
<code>\thickapprox</code> (\approx), 164	<code>\varepsilon</code> (ε), 158	<code>\zeta</code> (ζ), 158



索引

數學符號索引請見 393 頁。

- !‘ (¡), 67
- % (註銷指令), 67, 150, 293, 298, 324
- & (分隔欄位), 176, 191
- ’ (') (英文右單引號), 73
- ’’ (") (英文右雙引號), 73
- (, 177
- \((進入隨文數學模式), 154
- \[(進入展式數學模式), 155
- \$ (隨文數學模式), 154
- \$\$ (展式數學模式), 155
-), 177
- \) (離開隨文數學模式), 154
- \"o (ö 重音符號), 66
- \.o (õ 重音符號), 66
- /, 177
- ?‘ (¿), 67
- [, 177
- \ (TeX 指令), 68
- _ (空白指令), 70
- \# (#), 68
- \\$ (\$), 68
- \% (%), 67
- \& (&), 68
- \, (加入小空白), 73, 182
- \: (加大數學符號間距), 182
- \; (加大數學符號間距), 182
- | (界限符號), 177
- \\ (換行), 128
- * (換行), 128
- ˆ (數式上標), 68
- _ (數式下標), 68
- \{ (左大括號), 156, 177
- \\, 177
- \} (右大括號), 156, 177
- \] (離開展式數學模式), 155
-], 177
- \~{o} (ô 重音符號), 66
- \‘{o} (ò 重音符號), 66
- ‘ (') (英文左單引號), 73
- ‘‘ (") (英文左雙引號), 73
- 迷你版面指令環境, 137
- \~{o} (õ 重音符號), 66
- ~ (加入空白), 68, 189
- \! (縮小數學符號間距), 182
- \aa (â), 67
- \AA (Å), 67
- \abovecaptionskip , 206, 302
- abstract 指令環境, 89, 113
- \abstractname , 114, 284
- accents (重音符號), 66
- Acrobat Distiller, 361, 362
- Acrobat Reader, 39, 53, 263, 361
- \addcontentsline , 109, 116, 125, 335
- \address , 288, 291
- \addtocontents , 125
- \addtocounter , 285
- \addtolength , 98, 201
- Adobe Acrobat, 364
- Adobe Illustrator, 266
- \ae (æ), 67
- \AE (Æ), 67
- afm2tfm, 261, 272–274
- align 指令環境, 180
- align* 指令環境, 180, 182
- \alph , 117, 144
- \Alph , 117, 144
- \alsiname , 284
- $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, 153, 171
- $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$, 171
- amsmath 巨集套件, 159, 171, 179–182
- amssymb 巨集套件, 153, 162, 163
- amsthm 巨集套件, 171–175

- `\and` , 112
- answers 巨集套件, 307–309, 352
- `\appendix` , 126
- `\appendixname` , 126, 284
- `\arabic` , 117
- array 指令環境, 175, 186
- array 巨集套件, 104, 186, 190–202
- `\arraycolsep` , 199
- `\arrayrulecolor` , 217
- `\arrayrulewidth` , 199
- `\arraystretch` , 199
- Arseneau, Donald, 149, 212, 252
- article 文件類別, 40, 99
- `\atop` , 169
- `\author` , 25, 105, 112
- autoexec.bat , 59
- `\b{o}` (o 重音符號), 66
- babel 巨集套件, 66, 75
- `\backmatter` , 122, 349
- Barratt, Craig, 232
- Barroca, Leonor, 297
- baseline (基線), 82
- `\baselineskip` , 83, 98
- `\baselinestretch` , 83
- `\belowcaptionskip` , 206, 302
- Berry, Karl, 267, 269
- Bezos, Javier, 117, 323, 335
- `\bf` , 79, 80
- `\bfseries` , 79
- `\bibitem` , 105
- bibliography style (文獻資料格式), 311
- `\bibliographystyle` , 35
- `\bibname` , 284, 313
- bibtex, 21, 35, 311
- BiBTeX, 311
- `\big` , 170
- `\Big` , 170
- `\bigskip` , 130
- `\bline` , 216
- bm 巨集套件, 159
- `\bm` , 159
- `\bmod` , 167
- `\boldmath` , 159
- `\boldsymbol` , 159
- bookmarks, 364
- booktabs 巨集套件, 29, 186, 190, 194–195, 223
- `\bottomrule` , 194
- Bounding Box, 235
- box (文字方塊), 140
- `\bpara` , 214
- Braams, Johannes, 75
- Bringhurst, Robert, 74, 92, 259
- `\c{o}` (o 重音符號), 66
- `\caption` , 29, 124, 204–207, 213, 230, 242, 283 371
- caption2 巨集套件, 206, 222
- Carlisle, David, 85, 202, 217, 220, 223, 232, 253, 281
- `\cc` , 289
- `\ccname` , 284
- `\cdot` , 345
- center 指令環境, 129, 191, 230
- `\centering` , 128, 191, 327
- `\chapter` , 33, 89, 114, 326, 345
- `\chapter*` , 335
- `\chaptermark` , 344
- `\chaptername` , 118, 284, 344
- `\chk` , 282
- `\choose` , 169
- Christie, Agatha, 20
- `\circle` , 247, 248
- `\cite` , 105, 311
- `\citeauthor` , 319
- `\citep` , 35, 318
- `\citet` , 35, 318
- CJK 系統, 374
- `\cleardouble` , 122
- `\clearpage` , 122, 298, 348
- `\cline` , 195, 199
- `\closing` , 288
- cm (長度單位), 97
- `\cmidrule` , 194
- color 巨集套件, 217–220, 249–251, 253, 329
- `\colorbox` , 219, 250
- colortbl 巨集套件, 217–220
- `\columnbreak` , 152
- `\columncolor` , 217
- `\columnsep` , 99, 152
- `\columnseprule` , 99, 152
- `\columnwidth` , 99
- comment 指令環境, 150
- comment 巨集套件, 150
- config.sys , 61
- `\contentslabel` , 337
- `\contentsmargin` , 337
- `\contentsname` , 125, 284
- `\contentspage` , 337
- contour 巨集套件, 251

- `\contour` , 251
- `\copyright` (©), 65, 67, 342
- Corel Draw, 266
- `\count 0`, 122
- counter (計數器), 283
- crop 巨集套件, 358
- cropmark (裁切記號), 358
- cross-reference (交叉引述), 149–150, 184, 207–209, 362
- `\csp` (中文字距), 90, 196, 217
- `\csp` (中文字距), 90, 197, 330
- CTAN (Comprehensive T_EX Archive Network), 50, 96, 389
- `\ctxfbb` , 15, 25, 105, 111, 250, 260
- `\ctxfbbe` , 260
- `\ctxfbbes` , 260
- `\ctxfbbs` , 260
- `\ctxfdef` , 29, 33, 89, 105, 109, 119, 133, 146 344, 345
- `\ctxff` , 31
- `\ctxfk` , 25
- `\ctxfoff` , 370, 372
- `\ctxfon` , 370, 372
- `\ctxfr` , 29, 105
- `\ctxfvk` , 37
- `\ctxfvm` , 37
- `\cw` (cwT_EX), 278
- cwbibtex, 15, 21, 35, 311, 352
- cwmkidx, 356
- cwpt1, 373–379
- cwT_EX, 19, 20, 278
 - FreeBSD 版本, 64
 - Linux 版本安裝, 62–64 @@
 - OS2 版本, 64
 - Windows 與 Linux 版本差異, 63–64
 - 安裝, 54
 - 安裝問題, 58
 - 軟體下載, 389
 - 軟體更新, 55, 389
 - 網站, 50, 389
- cwtxex 巨集套件, 369
- cwtxex, 20, 40
 - + (選項), 296, 356
 - (選項), 296, 356
 - c (選項), 71
 - d (選項), 40
 - f (選項), 88
 - i (選項), 292, 370
 - l (選項), 90
 - s (選項), 88
 - z (選項), 90, 197
 - Z (選項), 90
 - zZ (選項), 90
- cwtxex.map , 376–379
- cwtxex1.map , 376–379
- cwttf, 373–379
- `\d{o}` (o 重音符號), 66
- `\dag` (†), 67
- Daly, Patrick W., 318, 321
- `\dashbox` , 248
- `\date` , 25, 27, 31, 105, 112
- dcolumn 巨集套件, 186, 193, 209–211
- `\ddag` (‡), 67
- `\def` , 277
- `\definecolor` , 217, 250, 329
- `\DefineShortVerb` , 149
- delimiter (界限符號), 176
- `\depth` , 140
- description 指令環境, 134–137
- Deutsch, L. Peter, 229
- displaymath 指令環境, 154
- `\displaystyle` , 183
- document 指令環境, 25, 33, 40, 96, 105
- `\documentclass`
 - article , 25
 - fleqno 選項, 103
 - fleqn 選項, 183
 - landscape 選項, 102
 - leqno 選項, 103
 - letter , 102, 287
 - oneside 選項, 110
 - openany 選項, 103, 110
 - report , 101
 - slides , 102
 - titlepage 選項, 103
 - twocolumn 選項, 103, 150
 - twoside 選項, 103
- `\documentclass` (文件類別指令), 25, 82, 96, 101, 104, 105
- `\documentstyle` , 104
- `\dotfill` , 132, 141
- `\dots` (...), 74
- `\doublerulesep` , 199
- Drakos, Nikos, 364
- drop caps, 251
- dropping shadows, 251
- Duggar, Angus, 274

- dvi (device independent), 47
- DVIPS, 41, 243, 261, 271–274, 362
- Eijkhout, Victor, 150
- em (長度單位), 97
- \em, 80
- em-dash (---), 74
- emTeX, 22, 39, 61, 268
- en-dash (--), 74
- Encapsulated PostScript (EPS), 229
- \encl, 289
- \enclname, 284
- \endfirsthead, 222
- \endfoot, 222
- \endhead, 222
- \endlastfoot, 222
- endnotes 巨集套件, 145
- \enlargethispage, 348
- enumerate 指令環境, 134–137, 282
- environment (指令環境), 89
- EPS (Encapsulated PostScript), 229
- eps2pdf, 306
- eqnarray 指令環境, 180
- eqnarray* 指令環境, 180
- equation 指令環境, 31, 154, 180
- \evensidemargin, 99
- ex (長度單位), 97
- Excel, 238
- \extrarowheight, 192, 199
- \fancyfoot, 341
- fancyhdr 巨集套件, 104, 110, 120–124, 323, 338–346
- \fancyhead, 341
- \fancypagestyle, 342
- fancyvrb 巨集套件, 146, 147
- \fbox, 140
- \fboxrule, 141
- \fboxsep, 141, 148, 219, 250
- Fear, Simon, 185, 194, 223
- figure 指令環境, 204–207, 230, 242, 302
- figure* 指令環境, 205
- \figurename, 206, 242, 284
- \fill, 131
- \filright, 334
- Flipo, Daniel, 251
- float (浮動版面), 186, 204–207
- \flushcolumns, 152
- flushleft 指令環境, 129
- flushright 指令環境, 129, 139
- \fnsymbol, 112, 144
- Foiltex 巨集套件, 300–307
- font, 77
 - Computer Modern, 77
 - outline font, 77
 - True Type, 78
 - Type 1 (PostScript), 77
- font family (字體族), 79
 - Roman family, 79
 - sans serif (無裝飾邊), 79
 - Typewriter family, 79
- font mapping file, 261, 270
- font metrics, 260
- font series (字體序列), 79
 - bold series, 79
 - medium series, 79
- font shape (字形), 79
 - italic, 79
 - slanted, 79
 - upright, 79
- font size (字級), 77, 78, 279
- \fontdimen 2, 260
- \fontfamily, 271, 291
- fontinst 巨集套件, 266–271
- Fontographer, 266, 380
- \fontseries, 271
- \fontshape, 271, 291
- \fontsize, 25, 40, 83, 85, 143, 271, 291
- footer (頁足), 101, 110
- footnote 計數器, 144
- \footnote, 15, 27, 143, 222, 284
- \footnotemark, 145
- \footnoterule, 144
- \footnotesep, 144
- \footnotesize, 82, 112
- \footnotetext, 145
- \footrulewidth, 340
- fpTeX, 20, 49, 61–62, 365
 - 網站, 390
- \framebox, 140
- Franz, Melchior, 358
- FreeBSD, 16, 64
- \frontmatter, 122, 349
- ftnright 巨集套件, 152
- gather 指令環境, 181
- Gauss, 240

- geometry 巨集套件, 37,
323–326, 352, 358
- Ghostscript, 52, 227–229, 361
- Grant, Michael C., 232
- graphics 巨集套件, 243
- graphicx 巨集套件, 27,
230–246, 253
- Ghostscript
印表機, 56
顯示密度, 56
- GSview, 39, 41, 52, 229, 361
- Guntermann, Klaus, 305
- Gustafson, Grant, 297
- Gutenberg, Johannes, 257
- \H{o} (ö 重音符號), 66
- Hafner, Jim, 300
- \hangindent , 278
- Harders, Harald, 251
- header (頁眉), 101, 110
- \headheight , 99
- \headpagename , 284
- \headrulewidth , 340
- \headtoname , 284
- \headwidth , 341
- Heath, Mark, 374
- \height , 140
- \hfill , 131, 141
- \hline , 191, 192, 199
- \hoffset , 101, 326
- \href , 363
- \hrulefill , 132, 333
- \hspace , 98, 130
- \hspace* , 98, 130
- htex386, 22, 387
- HTML (Hypertext Markup
Language), 359–372
- \htmladdnormallink , 372
- \htmlref , 370, 372
- \huge , 81
- \Huge , 81
- \hyperlink , 304, 363, 364
- hyperlink (超連結), 304, 368
- hyperref 巨集套件, 303, 304,
361–364
- \hypertarget , 304, 363, 364
- hyphen (-), 見「減號」, 74
- \ifodd , 122
- ifthen 巨集套件, 281
- \ifthenelse , 282
- \ignorespaces , 282
- in (長度單位), 97
- \include , 281, 295, 298, 349
- \includegraphics , 27,
229–242
- \includeonly , 295, 298, 349
- \indent , 127
- indent (行首內縮), 69
- \index , 353, 355
- \indexname , 284, 358
- \indexspace , 357
- initexmf, 261, 270, 294
- \input , 29, 280, 281, 309
- \intextsep , 253
- \isodd , 282
- \it , 79, 80
- \item , 134–137, 357
- itemize 指令環境, 134–137
- \itemsep , 136
- \itshape , 79
- Jakšić, Nikola, 306
- Jeffrey, Alan, 266
- \jobname , 67, 129, 340
- justification (右沿對齊), 128
- \kern , 90
- \kill , 188
- Knuth, Donald Ervin (高德
納), 19, 77, 153, 257
- Kuhlman, Volker, 321
- \l (l), 67
- \L (L), 67
- \label , 29, 149, 184, 207–209,
348, 371, 372
- \labelitemi , 135
- \labelitemii , 135
- \labelitemiii , 135
- \labelnumi , 135
- labels 巨集套件, 297–299
- Lamport, Leslie, 19
- landscape, 37, 102
- Lang, Russell, 229, 361
- \language , 75
- \leref , 107, 109, 278
- \large , 81
- \Large , 81
- \LARGE , 81, 327
- ℒ_{TeX}, 20, 277
ℒ_{TeX}2ε, 104
ℒ_{TeX}2.09, 104
- \LaTeX (ℒ_{TeX}), 65
- ℒ_{TeX}2HTML, 360, 364–372
- latexsym 巨集套件, 163, 166,
218
- Lavagnino, John, 145

- `\left` , 177
- `\left[` , 176
- `\lefteqn` , 181
- `\leftmargini` , 134, 136
- `\leftmarginii` , 134, 136
- `\leftmarginiii` , 134
- `\leftmark` , 124, 343, 346
- Leichter, Terry, 212
- Lemberg, Werner, 374
- `\let` , 196, 197
- letter 文件類別, 283
- lettrine 巨集套件, 251
- `\limits` , 161
- `\line` , 247, 248
- `\linespread` , 27, 83, 143, 300
- `\linewidth` , 99
- Linux, 15, 16, 49, 62–64
- list 指令環境, 134–137
- listbib 巨集套件, 321
- `\listfigurename` , 126, 284
- `\listoffigures` , 125, 126
- `\listoftables` , 125, 126
- `\listtablename` , 126, 284
- `\llap` , 見 `\rlap` , 211
- longtable 指令環境, 220–225
- longtable 巨集套件, 186, 220–225
- lscape 巨集套件, 220, 223
- `\LTbottomrule` , 223
- `\LTmidrule` , 223
- `\LTtoprule` , 223
- macros (巨集指令), 277–285
- `\mainmatter` , 122, 349
- `\makebox` , 140, 196
- makebst 巨集套件, 321
- makeidx 巨集套件, 355
- makeindex, 356
- `\makeindex` , 355
- `\maketitle` , 25, 105, 112, 341
- `\MaQ` (中文字體指令), 293
- `\marginpar` , 145, 146
- `\marginparpush` , 99, 146
- `\marginparsep` , 99, 146
- `\marginparwidth` , 99, 145
- `\markboth` , 344
- math 指令環境, 154
- math mode (數學模式), 154
- `\mathbf` , 159
- `\mathcal` , 158, 159
- Mathematica, 239
- `\mathindent` , 103, 183
- `\mathit` , 159
- mathematical environment (數學環境), 154
- mathpazo 巨集套件, 264
- `\mathrm` , 159
- `\mathsf` , 159
- `\mathtt` , 159
- `\mbox` , 122, 140, 178, 334
- McDonnell, Rowland, 266
- `\mdseries` , 79
- `\medskip` , 130, 277
- METAFONT, 77, 257–259
- `\midrule` , 194
- MiKTeX, 20, 39, 49, 365
 - 安裝, 53
 - 網站, 390
- minipage 指令環境, 137–141, 151, 241
- Mittelbach, Frank, 150, 171
- multicol 巨集套件, 37, 150–152
- multicolors 指令環境, 150–152
- `\multicolumn` , 150, 195, 199, 223
- `\multipt` , 248
- multirow 巨集套件, 212
- `\multirow` , 212
- multiline 指令環境, 180
- `\mutlicolsep` , 152
- mymacro 巨集套件, 84
- natbib 巨集套件, 35, 318–320
- netpbm, 365
- `\newcolumnmtype` , 210
- `\newcommand` , 277
- `\newenvironment` , 282
- `\newline` , 128
- `\newpage` , 109, 122, 129, 348
- `\newsavebox` , 142, 289
- `\newtheorem` , 171, 172, 174, 309
- `\newtheorem*` , 174
- `\newtheoremstyle` , 175
- NFSS (new font selection scheme), 78, 264
- `\nocite` , 319
- `\nocite*` , 319
- `\noindent` , 127
- `\nolimits` , 161
- `\nonumber` , 180
- `\normalfont` , 332
- `\normalsize` , 82
- `\not` , 163
- `\notag` , 180

- `\o` (ø), 67
- `\O` (Ø), 67
- `\oddsidemargin` , 99
- `odvips`, 22, 348
- `\oe` (œ), 67
- `\OE` (Œ), 67
- Omega, 22, 348
- `\onecolumn` , 351
- Oostrum, Piet van, 120, 212
- `\opening` , 288
- `\oval` , 247, 248
- `\P` (¶), 67
- package (巨集套件), 96, 103, 104
- page 計數器, 121
- page style (頁面格式), 101, 338
- `\pagebreak` , 152, 222
- `\pagecolor` , 250
- `\pagenumbering` , 121
- `\pageref` , 149, 207–209, 282
- `\pagestyle` , 121, 338, 339
 - empty 選項, 120
 - headings 選項, 120
 - myheadings 選項, 120
 - plain 選項, 120
- `\paperheight` , 99
- `\paperwidth` , 99
- `\par` , 69, 128, 129, 139
- `\paragraph` , 33, 114
- `\parbox` , 137–140, 241, 289
- `\parindent` , 127, 139, 173, 278
- `\parskip` , 128
- `\part` , 114
- `\partname` , 284
- Patashnik, Oren, 311
- path (搜尋執行檔), 58
- pause 巨集套件, 305–306
- `\pause` , 305
- pc (長度單位), 97
- PDF (Portable Document Format), 44, 229, 302–307, 359–372
- `pdfTeX`, 43, 45, 229, 302–307, 361
- `pdfscreen` 巨集套件, 300, 303–307
- perl, 365, 369
- picture 指令環境, 243, 246–249
- Piff, Mike, 295, 307
- `pltotf`, 268
- `\pmod` , 167
- Popineau, Fabrice, 49
- portrait, 102
- PostScript, 41, 42, 227–229
 - EPS, 229
- PostScript 字體, 257–272
- `\pounds` (£), 65, 67
- Power Point, 300, 302
- PPower4 巨集套件, 304–306
- preamble (全文設定區), 27, 83
- `\prefacename` , 284
- `\printindex` , 355
- printing points (點), 97
- `\proofname` , 173
- `\protect` , 207
- `\ps` , 289
- `ps2pdf`, 45, 361
- `ps2pk`, 380
- `ps2up`, 42
- `psbook`, 275
- `psfrag` 巨集套件, 27, 232–235, 306
- `\psfrag` , 27, 232, 233
- PSNFSS 巨集套件, 263
- `psnup`, 274, 275
- `pst-tree` 巨集套件, 253
- `pstcol` 巨集套件, 253
- PSTricks 巨集套件, 253–255
- `psutils`, 274–275
- pt (長度單位), 97
- Puga, Diego, 264
- `\put` , 247, 248
- `pxfonts` 巨集套件, 264
- `\qbezier` , 248
- `\qqquad` , 130
- `\quad` , 119, 130, 182
- quotation 指令環境, 133
- quote 指令環境, 89, 133
- Radhakrishnan, C.V., 303
- `\raggedcolumns` , 152
- `\raggedleft` , 128
- `\raggedright` , 128, 300, 334
- Rahtz, Sebastian, 263, 297, 361, 362
- `\raisebox` , 141
- Rancilio Audrey 咖啡機, 20
- `\rangle` , 216
- Reckdahl, Keith, 205, 207, 228
- `\ref` , 29, 149, 184, 207–209, 348, 371

- `\reflectbox` , 246
- `\refname` , 105, 284, 313
- `\renewcommand` , 29, 83, 105, 114, 277, 278
- `\renewenvironment` , 282
- `\resizebox` , 246
- `\resizebox*` , 246
- `\right` , 177
- `\right]` , 176
- `\rightmark` , 124, 343, 346
- `\rightskip` , 145
- `\rlap` , 見 `\llap` , 211, 216
- `\rm` , 79, 80
- `\rmfamily` , 79
- Rokicki, Tom, 272
- `\roman` , 117, 144
- `\Roman` , 117, 144, 331
- `\rotatebox` , 220, 223, 244
- `\rowcolor` , 217
- `\rule` , 98, 142, 144
- running-head, 115, 331
- Ryu, Young, 264
- `\S` (§) , 67, 119
- `\samepage` , 123
- sans serif (無裝飾邊字形), 見 serif, 92
- `\savebox` , 142
- `\sbox` , 289
- `\sc` , 79, 80
- `\scalebox` , 246
- Schöpf, Rainer, 171
- Schenk, Christian, 49
- `\scriptscriptstyle` , 183
- `\scriptsize` , 82
- `\scriptstyle` , 170, 183
- `\scshape` , 79
- `\section` , 33, 88, 105, 114, 326
- `\section*` , 116
- `\sectionmark` , 344, 345
- `\seename` , 284
- `\selectfont` , 271, 291
- seminar 文件類別, 300
- serif (裝飾邊字形), 見 serif, 92
- `\setcounter` , 121, 144, 184, 284
- `\setlength` , 83, 97
- `\sf` , 79, 80, 327
- `\sffamily` , 79, 332
- `\shortstack` , 247, 248
- `\signature` , 288
- `\sl` , 79, 80
- `\slshape` , 79
- `\small` , 27, 31, 82
- `\smallskip` , 130
- `\smallskipamount` , 130
- Sommerfeldt, Harald, 206
- `\special` , 227
- split 指令環境, 180
- `\sqrt` , 65, 157
- `\ss` (ß) , 67
- `\stackrel` , 169
- `\subitem` , 357
- `\subparagraph` , 33, 114
- `\subsection` , 33, 105, 114
- `\subsection*` , 116
- `\subsectionmark` , 344, 345
- `\subsubsection` , 33, 114
- `\swapnumber` , 174
- `\sz` , 12, 29
- `\t{oo}` (oo 重音符號), 66
- tabbing 指令環境, 186–190
- `\tabbingsep` , 189
- `\tabcolsep` , 192, 199
- table 指令環境, 29, 204–207, 242, 302
- table* 指令環境, 204
- `\tablename` , 125, 206, 283, 284
- tablenotes 指令環境, 213
- `\tableofcontents` , 109, 125, 335, 351
- tabular 指令環境, 29, 175, 186, 190–192, 202
- `\tabularcolumn` , 203
- tabularx 巨集套件, 186, 202–203
- `\tag` , 180
- `\tag*` , 180
- Taylor, Philip, 92
- `\tb` (排版迴歸式), 170, 279
- TeX, 19, 20, 277
 - 問題討論區, 389
 - 常見問題, 390
- TeX capacity exceeded (錯誤訊息), 383, 387
- tex2xtc, 125, 356, 371
- TeXLive, 49
- `\textbf` , 79
- `\textcolor` , 217, 220, 249–251, 253, 302, 329
- `\textheight` , 101, 323, 348
- `\textit` , 79
- `\textmd` , 79
- textmerg 巨集套件, 295–297

- `\textm` , 79
- `\textsc` , 79
- `\textsf` , 79
- `\textsl` , 79
- `\textstyle` , 183
- `\texttt` , 79
- `\textup` , 79
- `\textwidth` , 97, 101, 323, 341
- Thành, Hàn Thê, 361
- `\thanks` , 27, 112
- thebibliography 指令環境, 105, 311, 314, 320
- `\thechapter` , 309, 330, 331, 344
- `\thecontentslabel` , 337
- `\thefootnote` , 144
- theorem 巨集套件, 171–175, 301
- Theorem 巨集套件, 301
- `\theoremstyle` , 172, 175
- `\thepage` , 339
- `\thesection` , 116, 332, 345
- `\thesubsection` , 116
- `\thetitle` , 119
- `\thicklines` , 248
- `\thinlines` , 248
- `\thispagestyle` , 109, 121, 123, 341
- threeparttable 指令環境, 212
- threeparttable 巨集套件, 212–213, 220
- times 巨集套件, 263
- `\tiny` , 82
- `\title` , 15, 25, 31, 102, 105, 112
- `\titlecontents` , 336–338
- `\titlecontents*` , 337
- `\titleformat` , 326–335
- `\titlelabel` , 118
- `\titleline` , 327
- titlepage 指令環境, 111, 113
- `\titlerule` , 327
- titlesec 巨集套件, 107, 111, 116–119, 323, 326–335
- `\titlespacing` , 326–335
- titletoc 巨集套件, 335–338
- `\tnote` , 213
- tocdepth 計數器, 125
- `\today` , 66, 129, 289, 340
- `\topcaption` , 206
- `\topmargin` , 99
- `\toprule` , 194
- `\totalheight` , 140
- True Type 字體, 258
- `\tt` , 79, 80
- ttf2pfb, 272, 373
- ttf2pt1, 374
- `\ttfamily` , 79
- TUG (T_EX Users Group), 50, 389
- twocolumn 選項, 103
- `\twocolumn` , 351
- `\twoside` , 99
- txfonts 巨集套件, 31, 264
- type1cm 巨集套件, 85, 331
- `\u{o}` (ö 重音符號), 66
- Umeki, Hideo, 323
- `\unboldmath` , 159
- `\underline` , 80, 168
- unicode, 373
- `\unitlength` , 247
- `\upshape` , 79
- url 巨集套件, 149
- `\url` , 149
- `\usebox` , 142
- `\usepackage` , 27, 104
- `\v{o}` (ö 重音符號), 66
- van Oostrum, Piet, 338
- `\vector` , 247, 248
- `\verb` , 15, 146–149
- verbatim 指令環境, 15, 146–147
- Verbatim 指令環境, 146–149
- verse 指令環境, 133
- `\vfill` , 129, 131
- Visio 軟體, 238
- `\vline` , 199, 201
- `\voffset` , 101, 326
- vptovf, 268
- `\vrule` , 201, 203
- `\vspace` , 98, 130
- `\vspace*` , 98, 130
- Week, Andrew, 374
- `\width` , 140
- Windows, 49
- Windows Commander, 53
- Windvi, 39, 43, 44
- WinEdt, 39–41, 45–46, 52, 60, 198
- 中文標點符號, 72
- 功能鍵, 60
- 功能鍵設定, 45, 71
- 圖像 (icon) 設定, 43

- 輸入數學符號, 168
- 顯示密度, 56
- \wordsep , 331, 332
- wrapfig 巨集套件, 252
- wrapfigure 指令環境, 252
- YAP, 39, 43, 44
 - 中文字顯示問題, 44, 55
- \z (中文字距), 197, 293
- \Z (中文字距), 29, 209, 293
- \zZ (中文字距), 293
- Zandt, Timothy van, 146, 253, 300
- Zapf, Hermann, 264
- 3 劃
- 大宗信函, 295
- 工作檔案夾, 41, 55
- 4 劃
- 中文字距, 195–198
- 中文字體
 - 字體變形, 87, 259–264
- 中文字體指令, 86, 90
 - \ctxfdef , 29, 89, 119
 - 字距, 90, 197
 - 新式字體選用法, 86
 - 調整字距, 195–198
 - 調整垂直位置, 90
- 分式, 156
- 文件類別 (document class)
 - article , 104–107
- 文件類別 (document class), 101–107
 - amsart , 102
 - amsbook , 102
 - article , 40, 102
 - book , 33, 110–114
 - foils , 300
 - letter , 102, 287, 289
 - report , 102
 - slides , 102
- 文件類別指令
 - (\documentclass), 96
 - twocolumn 選項, 122
- 文字方塊 (box), 140
 - LR 方塊, 140
 - 段落方塊, 140
 - 線條方塊, 140
- 文稿結構 (document structure), 95, 101–107
- 文獻資料格式, 311
- 巨集指令 (macros), 277–285
 - mymacro.tex , 29
 - 中文, 281
- 巨集套件, 96, 103–104
 - amsmath, 159, 171, 179–182
 - amssymb, 153, 162, 163
 - amsthm, 171–175
 - answers, 307–309, 352
 - array, 104, 186, 190–202
 - babel, 66, 75
 - bm, 159
 - booktabs, 29, 186, 190, 194–195, 223
 - caption2, 206, 222
 - color, 217–220, 249–251, 253, 329
 - colortbl, 217–220
 - comment, 150
 - contour, 251
 - crop, 358
 - cwtex, 369
 - dcolumn, 186, 193, 209–211
 - endnotes, 145
 - fancyhdr, 104, 110, 120–124, 323, 338–346
 - fancyvrb, 146, 147
 - Foiltex, 300–307
 - fontinst, 266–271
 - ftnright, 152
 - geometry, 37, 323–326, 352, 358
 - graphics, 243
 - graphicx, 27, 230–246, 253
 - hyperref, 303, 304, 361–364
 - ifthen, 281
 - labels, 297–299
 - latexsym, 163, 166, 218
 - lettrine, 251
 - listbib, 321
 - longtable, 186, 220–225
 - lscap, 220, 223
 - makebst, 321
 - makeidx, 355
 - mathpazo, 264
 - multicol, 37, 150–152
 - multirow, 212
 - mymacro, 84
 - natbib, 35, 318–320
 - pause, 305–306
 - pdfscreen, 300, 303–307

- PPower4, 304–306
- psfrag, 27, 232–235, 306
- PSNFSS, 263
- pst-tree, 253
- pstcol, 253
- PSTricks, 253–255
- pxfonts, 264
- tabularx, 186, 202–203
- textmerg, 295–297
- theorem, 171–175, 301
- Theorem, 301
- threeparttable, 212–213, 220
- times, 263
- titlesec, 107, 111, 116, 117, 323, 326–335
- titletoc, 335–338
- txfonts, 31, 264
- type1cm, 85, 331
- url, 149
- wrapfig, 252

- 5 劃
- 正文方格, 99
- 目錄, 124, 335
- 立體陰影 (dropping shadows), 251

- 6 劃
- 交叉引述 (cross-reference), 149–150, 184, 207–209, 348, 362
- \label, 149, 184, 207–209
- \pageref, 149
- \ref, 149, 184, 207–209

- 數學式, 184
- 全文設定區 (preamble), 27, 83, 99
- 多欄位版面, 103, 150–152
- 字元排序 (encoding), 263
- 字型
 - 描點字型, 44
 - 字型度量 (font metrics), 260
 - 字型度量檔, 266
 - 字型規格檔, 60, 380
 - 字型對應檔, 376
 - 字型對應檔 (font mapping file), 261, 269, 270
 - 字級 (font size), 77, 78, 279
 - 字體, 77
 - Charter, 264
 - Computer Modern, 31, 257
 - PostScript 格式, 259
 - Garamond, 263, 264
 - Helvetica, 264, 269
 - Mathtime, 264
 - METAFONT, 257
 - outline font, 77
 - Palatino, 264, 266
 - PostScript, 257–272
 - small capital, 271
 - Times New Roman, 263
 - Times Roman, 31, 264
 - True Type, 45, 78, 259, 272, 373
 - Type 1 (PostScript), 77, 78
 - unicode, 373
 - Utopia, 264
 - Windows 字體, 375
 - 中文字排序, 379
- 中文字距
 - \csp, 90
 - \cspp, 90
 - \kern, 90
- 中文字體, 259
- 中文字體名稱, 378
- 字形 (font shape), 79
- 字級, 81
- 字體序列 (font series), 79
- 字體族 (font family), 79
- 設計尺寸 (design size), 81, 97
- 造字, 380
- 描點字型, 258
- 描邊字型, 258
- 數學字體, 159, 259, 264
- 數學粗體, 159, 302
- 字體指令, 271–272
 - \ctxfoff, 370
 - \ctxfon, 370
 - NFSS, 15
 - 宣告字體指令, 79
 - 相對大小, 81
 - \footnotesize, 82
 - \huge, 81
 - \Huge, 81
 - \large, 81
 - \Large, 81
 - \LARGE, 81
 - \normalsize, 82
 - \scriptsize, 82
 - \small, 82
 - \tiny, 82
 - 數學字體指令, 159
 - 標準字體指令, 79

- 灰階 (grayscale), 249
- 行列式, 175–179
- 行首內縮 (indent), 69, 127
- 行距, 82, 90
 - `\linespread` , 300
- 行寬, 91, 99
- 7 劃
- 希臘字母符號, 158
- 投影片, 15, 101, 299–307
 - PDF, 302
 - 彩色, 302
- 李君宇, 374
- 李果正, 62, 374
- 8 劃
- 函數符號, 160
- 固定長度 (fixed length), 見
 - 「彈性長度」, 97
- 固定格式標籤, 297–299
- 居中編排, 128
- 放大或縮小圖表, 245
- 版面
 - 直向 (portrait), 102
 - 橫向 (landscape), 102
- 表格
 - array 巨集套件, 186, 190–202
 - array 指令環境, 186
 - booktabs 巨集套件, 186, 190, 194
 - dcolumn 巨集套件, 186
 - tabbing 指令環境, 186–190
 - tabular 指令環境, 186
 - tabularx 巨集套件, 186
 - tabularx 指令環境, 202–203
 - 小數點對齊, 209
 - 加大行距, 192
 - 居中, 191
 - 斜線, 214–216
 - 旋轉, 220
 - 短直線, 201
 - 註解, 199, 212–213
 - 超大型, 220–225
 - 數字上下對齊, 193
 - 標題, 204–207
 - 欄位寬度, 202
 - 長度單位, 97
 - cm, 97
 - em, 97
 - ex, 97
 - in, 97
 - pc (pica), 97
 - pt (point), 97
 - 附錄, 126
 - 9 劃
 - 信函, 101, 287–299
 - 大宗信函, 295–297
 - 中文信頭標誌, 291
 - 信頭標誌, 289–297
 - 指令 (command)
 - 強制變數 (mandatory argument), 98
 - 選項變數 (optional argument), 98
 - 指令環境 (environment), 89
 - 段落間距 (`\parskip`), 128
 - 界限符號 (delimiter), 176, 177, 216
 - 美國數學學會, 153, 161
 - 計數器 (counter), 283
 - addtocounter , 285
 - chapter , 284, 331
 - enumi , 284
 - enumii , 284
 - enumiii , 284
 - enumvi , 284
 - equation , 184, 284
 - figure , 284
 - footnote , 144, 284
 - mpfootnote , 284
 - page , 121, 284
 - paragraph , 284
 - part , 284
 - section , 116, 284
 - subparagraph , 284
 - subsection , 116, 284
 - subsubsection , 284
 - table , 284
 - 重音符號 (accents), 66
 - 頁足 (footer), 101, 110, 123, 342–346
 - 頁眉 (header), 99, 101, 110, 123, 342–346
 - 頁眉之章節標題, 124
 - 中文字亂碼問題, 120, 123
 - 頁面格式 (page style), 101, 120, 338
 - 頁碼, 103, 120–122, 338–346, 349
 - 頁碼字體, 121
 - 頁碼計數器, 121

- 10 劃
- 展示數式 (display formula),
154
- 浮動版面 (float), 186, 204, 242
figure 指令環境, 186, 204
table 指令環境, 186, 204
- 矩陣, 175–179
- 索引, 283, 352–358
特殊符號, 354
排序, 358
標註索引名詞, 353
- 紙張尺寸, 99
a4paper , 99
a5paper , 99
b5paper , 99
executivepaper , 99
legalpaper , 99
letterpaper , 99
- 翁鴻翎, 15, 62, 311, 374
- 迷你版面, 137–143
- 迴歸式, 170, 279
- 記憶體不足, 348
- 11 劃
- 參考文獻, 21, 35, 105, 109,
311–322, 352
中文排序, 315
文獻格式, 321
文獻資料檔, 311, 312,
314–317
- 基線 (baseline), 82, 138, 176
- 彩色圖文, 249–251
- 排版, 66
日文, 29, 212
多欄, 150–152
- 兩欄, 127, 150–152, 351
- 法文, 75
- 直排, 37, 87
- 研究生論文, 107–110
- 書籍, 349–353
- 單欄, 351
- 幕前排版, 39
- 幕後排版, 39
- 德文, 66, 75
- 橫排, 37, 87
- 排版步驟, 40–45
- 排版訊息, 46
- 旋轉文字或圖表, 244
- 條列指令環境, 134–137
description , 134
enumerate , 134
itemize , 134
- 條列項內縮距離
(\leftmargini), 136
- 習題與解答, 307–309
- 造字, 373–381
- 章節標題, 114–120, 323, 326
彩色, 329
- 12 劃
- 換行指令
\\, 112, 128
\newline , 128
- 換頁, 122
- 期望值指令, 160
- 減號 (—), 見 hyphen, 74
- 裁切記號 (cropmark), 358
- 註解, 143–146
\fnsymbol , 144
- 分隔線 (\footnoterule),
144
- 間距 (\footnotesep), 144
- 註銷指令 (%), 67
- 超連結 (hyperlink), 304, 368
- 13 劃
- 填入直線 (\hrulefill), 132
- 填入細點 (\dotfill), 132
- 微分符號, 156
- 照列原文 (verbatim), 146–149
- 14 劃
- 圖形
- bounding box, 273
- Corel Draw, 238
- EPS, 229, 235–242, 273
- EPS Bounding Box, 235
- Excel, 238
- Gauss, 240
- PostScript 印表機驅動程
式, 236
- Visio, 238
- 中文標誌, 231–234
- 引用外製圖形, 229–235
- 列印品質, 236
- 印表機, 228
PostScript 印表機, 229
- 放大或縮小, 245
- 常見問題, 240
- 彩色, 249–251
- 旋轉, 244
- 描點圖形, 227, 241
- 描邊圖形, 227
- 圖檔規格, 227–229
- 數學式, 231–234

- 標題, 230, 242
- 調整大小, 230
- 樹狀圖, 248
- 轉換格式, 235, 237, 241, 306
- 繪圖軟體, 237–241
- 顯示問題, 230, 235, 239, 240, 243, 249
- 圖表
 - 標題, 205
- 圖檔規格
 - EPS, 45, 229–242
 - JPEG, 45, 229
 - PDF, 45, 229
- 摘要, 113
- 網路出版, 359–372
- 15 劃
- 彈性長度 (rubber length), 見「固定長度」, 97
- 數字格式
 - alph, 121
 - \alph, 117
 - Alph, 121
 - \Alph, 117
 - arabic, 121
 - \arabic, 117
 - roman, 121
 - \roman, 117
 - Roman, 121
 - \Roman, 117
- 數學式
 - 下標, 156
 - 上標, 156
 - 加總函數, 160
 - 左下標, 178
 - 交叉引述, 184
 - 多行數學式, 179–182
 - 定理, 171
 - 定義, 171
 - 定義與定理, 171–175
 - 居中或靠左, 183
 - 界限符號, 177
 - 相對關係符號, 161
 - 矩陣與行列式, 175–179
 - 迴歸式, 170
 - 粗體符號, 159
 - 連分式, 183
 - 畫底線, 168
 - 證明, 172
 - 開根號, 157
 - 間距調整, 182
 - 數學字體指令, 159
 - 數學花體字 (script letters), 158
 - 箭號 (arrow), 161
 - 編號, 182
 - 調整符號大小
 - \displaystyle, 183
 - \scriptscriptstyle, 183
 - \scriptstyle, 183
 - \textstyle, 183
 - 調整間距
 - \!, 182
 - \,, 182
 - \:, 182
 - \;, 182
 - 積分符號, 160
 - 雙元運算符號, 165
- 數式內排版中文字, 178
- 數式內排版普通文字, 178
- 數學重音符號, 168
- 數學符號間距調整, 182
- 數學模式, 154
- 數學環境, 154
- 標點符號
 - 中文標點符號, 71, 72
 - 英文右單引號 (’), 73
 - 英文右雙引號 (”), 73
 - 英文左單引號 (‘), 73
 - 英文左雙引號 (“), 73
 - 英文標點符號, 71
 - 破折號, 73
 - 避頭點, 93
- 標題, 283
 - 中文字, 283
 - 序文與索引, 335
 - 章節, 323, 326, 340, 351
 - 圖表, 230
 - 標題頁格式, 340
- 標題格式
 - block, 328
 - display, 327, 328, 330
 - drop, 328
 - frame, 328
 - hang, 327, 330
 - leftmargin, 328
 - rightmargin, 328
 - runin, 327
 - wrap, 328
- 線條方塊 (rule), 142
- 調整間距, 130
- 靠右編排 (flushright), 128
- 靠左編排 (flushleft), 128

樹狀圖 (trees), [254–255](#)

16 劃

輸入文稿, [74](#)

中文稿, [70–74](#)

半型中文, [72](#)

全型中文, [72](#)

英文稿, [68–70](#)

數學文稿, [155](#)

標點符號, [71](#)

錯誤訊息, [383](#)

隨文數式 (in-text formula),
[154](#)

17–23 劃

儲存方塊 (\sbox), [142](#)

檔案搜尋路徑, [57, 62](#)

避頭點, [93](#)

點 (printing point), [97](#)

邊註 (marginal notes), [99,](#)
[143, 145, 146](#)

變異數, [160](#)