

cw T_EX 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)

吳聰敏·吳聰慧

CTW *TEX*
第3版 排版系統

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 cwtex 中文處理程式, 23 套橫排與 5 套直排中文 Type 1 字型檔。以上軟體之版權屬吳聰敏與吳聰慧所有。此外, 光碟內 5 套中文 TrueType 字型檔及 cwbibtex 程式 (初版) 之版權為翁鴻翎所有。上列軟體中, 排版程式以及 5 套橫排 Type 1 及 TrueType 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 \LaTeX Project Public License 之條件。詳細說明, 請見 CTAN: $+\text{macros/latex/base/lppl.txt}+$ 。

©2005

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

台北市重慶南路 1 段 121 號 5 樓之 11

電話: (02)2382-1120, 2382-1169

傳真: (02)2331-4416

郵撥: 15718419

Email: hanlu@hanlu.com.tw

$+\text{http://www.hanlu.com.tw}+$

1997 年 10 月初版

2005 年 10 月 3 版

定價: 新台幣 600 元

cwTeX 排版系統 / 吳聰敏·吳聰慧 [著]. — 3 版. —

臺北市: 吳聰敏出版: 翰蘆圖書總經銷, 2005 [民 94]

面; 公分

參考書目: 面

ISBN 957-41-3163-7 (平裝附光碟片)

1. TEX (電腦程式)

312.949T46

94019070

目錄

3 版序	15
1 前言	17
1.1 T _E X 與 Word 不同	17
1.2 L ^A T _E X 只是一個排版工具	18
1.3 本書章節	19
2 例子	21
3 排版方法	37
3.1 檔案管理	37
3.1.1 DOS 視窗	38
3.1.2 批次檔	38
3.2 排版之步驟	39
3.3 文字編輯軟體操作介面	40
3.3.1 WinEdt 操作介面	40
3.3.2 排版英文稿	42
3.3.3 Crimson editor 操作界面	42
3.3.4 工作子目錄	43
3.4 編輯功能鍵之設定	44
3.4.1 輸入中文標點符號	45
4 安裝與設定	47
4.1 安裝	47
4.2 測試	49
4.2.1 常見之安裝問題	49
4.3 手動設定	51

4.3.1	文字編輯軟體	51
4.3.2	Total commander 檔案管理	52
4.3.3	檔案搜尋	52
4.4	軟體更新	53
4.4.1	加裝字型	54
4.4.2	安裝其他軟體	54
4.5	Linux 版本	54
5	指令與文字	55
5.1	符號與指令	55
5.1.1	註銷指令	57
5.1.2	排版網址	58
5.2	英文稿件輸入原則	59
5.3	中文稿件輸入原則	60
5.3.1	標點符號	62
5.3.2	英文引號與破折號	63
5.3.3	避頭點	65
5.4	排版其他西方國家文字	65
6	選用字體	67
6.1	字體規格	67
6.2	選用字體	68
6.2.1	英文與數字字體	68
6.2.2	中文字體	71
6.2.3	設定全文之中文字體	73
6.3	改變字級	74
6.3.1	相對大小之字級指令	75
6.3.2	行距	76
6.3.3	同時變更字級與行距	76
6.3.4	選用任意點數之字體	77
6.4	選擇字級與行距	78
7	文稿結構	81
7.1	長度單位	82
7.1.1	版面大小	83

7.2	文稿結構	86
7.2.1	指令選項	86
7.2.2	巨集套件	87
7.3	文稿類別	88
7.3.1	article 文稿類別	89
7.3.2	book 文稿類別	92
7.4	題目與摘要	94
7.4.1	摘要	96
7.5	章節標題	96
7.5.1	章節標題之層級	96
7.5.2	設定章節標題之字體	98
7.5.3	titlesec 巨集套件	99
7.5.4	中文標題之字體	100
7.6	引述章節	101
7.7	頁面格式與頁碼	102
7.7.1	頁碼	103
7.7.2	換頁	104
7.8	目錄	106
7.9	附錄	107
8	段落編排	109
8.1	段落格式	109
8.1.1	居中與靠邊	110
8.2	調整間距	111
8.2.1	插入空白	111
8.2.2	填入細點或直線	113
8.3	引文與詩詞	114
8.3.1	verse 指令環境	114
8.3.2	quote 指令環境	114
8.4	條列指令環境	115
8.5	迷你版面與方塊	117
8.5.1	迷你版面	118
8.5.2	方塊	121

8.5.3	線條方塊	122
8.5.4	儲存方塊	123
8.6	註解與邊註	123
8.6.1	註解	123
8.6.2	註解編號	124
8.6.3	邊註	125
8.7	多欄位版面	126
8.8	comment 巨集套件	128
8.9	照列原文	128
8.9.1	Verbatim 指令環境	129
9	數學式子	131
9.1	科技文稿之排版規範	131
9.2	數式環境	132
9.2.1	數學文稿輸入原則	133
9.2.2	運算符號與上下標	134
9.2.3	分式	134
9.2.4	開根號	135
9.3	數學符號	135
9.3.1	希臘字母符號	135
9.3.2	粗體數學符號	136
9.3.3	函數符號	137
9.3.4	積分與加總函數	138
9.3.5	箭頭符號	139
9.3.6	相對關係符號	140
9.3.7	雙元運算符號	143
9.3.8	數學重音符號	145
9.3.9	上下重疊符號	147
9.3.10	連續點	148
9.4	矩陣與行列式	148
9.4.1	界限符號	149
9.5	定義與定理	152
9.5.1	amsthm 巨集套件	152

9.5.2	定理編號與排版格式	154
9.5.3	定理證明	154
9.6	amsmath 巨集套件	155
9.6.1	引述數式	158
9.7	細節調整	158
9.7.1	調整符號間距與大小	159
9.7.2	數學式居中與靠左	160
10	表格	161
10.1	tabular 指令環境	162
10.1.1	控制欄位間距	165
10.1.2	booktabs 巨集套件	166
10.1.3	表格內的文字段落	167
10.1.4	表格註解	169
10.1.5	array 巨集套件	169
10.2	tabularx 巨集套件	173
10.3	圖表位置與標題	175
10.3.1	浮動版面指令環境	175
10.3.2	圖表標題	176
10.3.3	caption 巨集套件	177
10.4	引述表格	178
10.5	表格排版細節調整	179
10.5.1	表格小數點對齊	179
10.5.2	橫列文字對齊	182
10.5.3	表格內加入括弧或斜線	182
10.6	彩色表格	184
10.7	超大型表格	187
10.7.1	旋轉表格	187
10.7.2	超長表格	188
10.8	tabbing 指令環境	193
10.8.1	以樣本行設定距離	194
10.8.2	其他控制指令	195

11 引用外製圖形	197
11.1 圖形檔案規格	197
11.2 引用 PostScript 圖形	198
11.2.1 graphicx 巨集套件	199
11.2.2 圖形內加入中文或數式	201
11.2.3 psfrag 巨集套件	202
11.3 輸出 EPS 圖形	204
11.3.1 PostScript 印表機驅動程式	204
11.3.2 使用繪圖軟體繪製 EPS 圖形	206
11.4 引用描點圖形	208
11.5 引用 PDF 圖形	209
12 圖形與彩色	213
12.1 旋轉與縮放文字圖表	213
12.1.1 旋轉文字圖表	214
12.1.2 縮放文字圖表	215
12.2 彩色圖文	216
12.2.1 選擇色彩	218
12.3 圖形小工具	219
12.3.1 lettrine 巨集套件	219
12.3.2 wrapfig 巨集套件	220
12.3.3 contour 巨集套件	221
12.4 picture 指令環境	221
12.5 PSTricks 巨集套件	224
13 投影片與信函	227
13.1 投影片	227
13.1.1 beamer 巨集套件	227
13.1.2 動態呈現	229
13.1.3 投影片版面設計	231
13.2 信函	232
13.2.1 letter 文稿類別	233
13.2.2 設計信頭標識	234
13.3 大宗信函	236

13.4	固定格式標籤	239
13.5	考題與解答	241
14	巨集指令	245
14.1	定義巨集指令	245
14.1.1	設定字級之巨集指令	247
14.1.2	巨集指令與中文	248
14.1.3	依條件處理之巨集指令	249
14.2	定義指令環境	250
14.3	計數器	251
14.4	中文標題	252
15	參考文獻	253
15.1	排版參考文獻	253
15.1.1	設定選項	255
15.1.2	搜尋文獻資料檔	256
15.2	建立文獻資料檔	256
15.2.1	cwbibtex 訊息	258
15.2.2	文獻資料檔格式	259
15.3	natbib 巨集套件	262
15.3.1	徵引文獻之指令	262
15.4	bibentry 巨集套件	264
15.5	參考文獻工具軟體	264
15.5.1	Jabref 軟體	265
15.5.2	製作文獻格式檔	265
16	PostScript 字體	267
16.1	英文 Type 1 字體	267
16.1.1	PSNFSS 巨集套件	268
16.2	Metafont 字體	270
16.2.1	描邊字型與描點字型	270
16.2.2	PostScript 格式之 CM 字體	270
16.3	中文 Type 1 字體	272
16.3.1	中文變形字	273
16.3.2	字型對應檔	274

16.4	創造英文 Type 1 字體巨集套件	275
16.4.1	fontinst 巨集套件	275
16.4.2	選用字體之指令	279
16.4.3	TrueType 字型	281
17	版面設計	283
17.1	geometry 巨集套件	283
17.2	chnpage 巨集套件	286
17.3	titlesec 巨集套件進階指令	286
17.3.1	彩色標題	289
17.3.2	章節計數器	292
17.3.3	節標題之設計	292
17.3.4	嵌入標題	293
17.3.5	序文與索引之標題	294
17.4	titletoc 巨集套件	295
17.5	頁面格式	298
17.5.1	頁眉之章節標題	299
17.5.2	fancyhdr 巨集套件	302
17.5.3	單雙頁版面不同	303
18	索引	305
18.1	標識索引名詞	305
18.1.1	特殊符號	307
18.2	排版索引的步驟	308
18.2.1	排版索引之訊息	311
19	書籍排版	313
19.1	排版設計	313
19.2	書籍排版例子	314
19.3	完稿小工具	319
19.3.1	裁切記號	319
19.3.2	dvips 程式	320
19.3.3	psutils 工具程式	320

20	網路出版	323
20.1	HTML 與 PDF 的比較	323
20.2	輸出 PDF 檔案	324
20.2.1	hyperref 巨集套件	326
20.2.2	PDF 書籤	327
20.3	TeX2HTML	328
20.3.1	安裝	328
20.3.2	測試與更新	329
20.3.3	排版方法	329
20.3.4	排版測試檔	330
21	造字	335
21.1	造新字	335
21.2	由 TrueType 輸出 cwTeX 字型檔	337
21.2.1	cwpt1 與 cwttf 軟體	337
21.2.2	轉換方法	338
21.2.3	更改設定	340
21.3	專業造字軟體	341
21.3.1	使用新字型	341
21.3.2	中文字之排序	342
22	排版訊息	343
22.1	排版訊息	343
22.2	cwtex 訊息	345
22.3	latex 或 pdflatex 訊息	346
23	cwTeX 系統	349
23.1	中文轉換程式與字型檔	349
23.2	舊式字體指令	350
23.2.1	調整中文字距	351
	參考書目	353
	數學符號索引	355
	索引	361

3 版序

cwTeX 使用手冊初版於1997年出版，當時的排版程式是以 DOS 系統下之 emTeX 為基礎。第2版出版於2000年，書內介紹的是 Windows 作業系統下的 MiKTeX 與 fpTeX 兩套系統。不過，當時的 latex 程式無法排版整本手冊（約350頁），dvips 程式要輸出整本手冊也會出問題。我們必須藉助其他程式解決。2002年出版修訂2版時，只要經過適當設定，latex 程式可以毫無困難地排版整本手冊，但是 dvips 仍有困難。此次排版第3版的手冊時，我們所使用的 MiKTeX 可以輕鬆地排出整本手冊，dvips 輸出時也不會出問題。

以上的轉變反映電腦硬體與軟體的進步。軟體的進步也改變了電腦使用者的習慣。在第1版手冊內，我們的說明中提及 DOS, Win31, 及 Win95。但今天初學電腦的人大多已不熟悉 DOS 指令了。不過，TeX 程式仍然是在 DOS 視窗內執行，故排版者對於 DOS 指令仍然必須有簡單的了解。

TeX 排版系統發展於1980年代。以電腦軟體而言，這幾乎已是考古學的對象了。今天我們不容易找到幾個經過20幾年的市場競爭，而仍存活著的軟體。幸運的是，TeX 系統的基本設計讓它能隨著電腦科技的進步而進展。譬如，在20年前，PDF 檔案格式尚未出現，但今天我們可以利用 L^AT_EX 排版出能搜尋中文字詞的 PDF。我們也可以排版出以單槍投影機放映之投影片。

科技進步的確讓電腦排版更為方便。不過，對於初學者而言，使用排版軟體的主要問題仍然是軟體安裝是否容易？使用手冊是否正確？清楚？易讀？cwTeX 並非我們的本業，不過，多年來使用 L^AT_EX 排版文稿，以及與讀者的意見交流，我們累積了不少經驗，這些經驗反映在本版的大幅度修正上。拿本版與修訂2版比較一下，你馬上就會注意到本版修正幅度之大。

就軟體而言，本版的主要更新如下：

- 安裝更容易
- PDF 檔案可搜尋中文字詞

- cwbibtex 參考文獻工具程式功能增強
- cwmkidx 索引程式功能增強
- 提供 makekeys 程式以輸出 EPS 圖形檔
- 簡易造字 (新字圖形)
- 可由 TrueType 字型檔輸出 cwTeX 字型檔
- 提供 5 套 TrueType 字型, 可使用於 Windows 應用軟體內
- 提供 \TeX 2HTML 安裝程式
- cwTeX 簡體字版本 (含 3 套字型檔)

在修訂 2 版中, 我們較詳細地說明 Linux 版本之安裝方法。本版主要是更新 Windows 版本之程式, 為了節省篇幅, Linux 程式之安裝說明直接置於網頁上。有意使用者, 請進入 cwTeX 之 ftp 網址, 再進入 \cwTeX_linux, 參閱其中之 README。

cwTeX 另有排版簡體字之版本, 內含 3 套字型檔。有意測試使用者, 請由上述網址之 \cwTeX-GB 檔案夾下載。

就手冊內容而言, 我們認為本版比起舊版而言, 解釋更清楚, 文字更易讀。我們不敢說內容沒有錯誤, 但應該是大幅減少。在累積了多年的使用經驗之後, 我們更了解如何才能讓初學者很快地掌握 \TeX 排版之原理與技巧。不過, 我們也期待讀者如以往一樣, 提出批評與改進意見。

本版準備過程中, 我們特別感謝盧佳慧小姐細心校對及檢驗書中所有的例子。陳俞成老師在本文初稿中找出許多錯誤, 陳旭昇老師 (間接地) 建議幾個更新的方向。吳慕凡小姐為本書製作小插圖, 我們謹在此表達感謝之意。

吳聰敏·吳聰慧

2005.10

1 前言

爲什麼要使用 $\text{T}_\text{E}\text{X}$ 排版? 你的答案可能是: 老闆 (指導教授) 要求。如果你不幸是屬於這一群, 解決問題的方法是換個老闆。換了老闆之後, 你可能還是有興趣了解: 爲什麼那個老傢伙要使用 $\text{T}_\text{E}\text{X}$? 你如果問 Donald E. Knuth (高德納) 這個問題, 他的回答是:¹

“[T]he $\text{T}_\text{E}\text{X}$ 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, $\text{T}_\text{E}\text{X}$ 的真正精神是排版品質, 至少 Knuth 是這麼認爲。但如果你相信這個答案, 有人會告訴你: 「 $\text{T}_\text{E}\text{X}$ 是 Knuth 發明的, 你相信他講的?」

本書說明如何使用 $\text{T}_\text{E}\text{X}$ 排版, 因此, 我首先應該說明, **Word** 又流行又好用, 爲何要使用 $\text{T}_\text{E}\text{X}$? 但我是 $\text{cW}_\text{E}\text{X}$ 的作者之一,² 而 $\text{cW}_\text{E}\text{X}$ 是建立在 $\text{T}_\text{E}\text{X}$ 排版系統上。因此, 我所說的任何關於 $\text{T}_\text{E}\text{X}$ 或 $\text{cW}_\text{E}\text{X}$ 的好處, 你都不應該相信, 或者至少是半信半疑。因此這個問題還是讓你自己找答案吧!

1.1 $\text{T}_\text{E}\text{X}$ 與 **Word** 不同

$\text{T}_\text{E}\text{X}$ 與 **Word** 不同。**Word** 是 MicroSoft 的產品, 替 Bill Gates 賺了不少錢。 $\text{T}_\text{E}\text{X}$ 是 Donald E. Knuth 所寫, 但它是 freeware, 免費提供使用。Knuth 可能因爲賣 $\text{T}_\text{E}\text{X}$ 手冊而賺了一點錢, 但與 Bill Gates 所賺的錢是不能相比的。不過, 誰賺的錢多與本書無關。 $\text{T}_\text{E}\text{X}$ 與 **Word** 不同的地方在於, **Word** 是幕前排版, $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 則是幕後排

¹見 Knuth (1999), 頁 559, 616。

² $\text{cW}_\text{E}\text{X}$ 是由吳聰敏·吳聰慧·翁鴻翎共同發展。

版。使用 $\text{T}_\text{E}\text{X}$ 排版，你需要有一個文字編輯軟體，先把所有文字與指令輸入在一個檔案內，再對此檔案執行 tex 程式以排版出結果來。

$\text{L}\text{A}\text{T}_\text{E}\text{X}$ 是從 $\text{T}_\text{E}\text{X}$ 衍生出來的。簡單來說， $\text{T}_\text{E}\text{X}$ 之指令較為繁複； $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 的指令較為簡單，容易上手。而且， $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 系統有專家持續作維護與更新，因此能跟得上排版科技的發展。譬如，單槍投影機普及之後， $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 也有好幾套工具發展出來，以排版單槍投影機使用之投影片。本書主要介紹 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ ，不過行文當中我們有時候會把 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 與 $\text{T}_\text{E}\text{X}$ 兩個名詞混著使用。

$\text{c}\text{w}\text{T}_\text{E}\text{X}$ 的功能是將中文字轉換成 $\text{T}_\text{E}\text{X}$ 格式。它本身並無排版的能力，只是把文稿內之中文字轉換成 $\text{T}_\text{E}\text{X}$ 的格式，再交由 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 排版。在一般的電腦術語中， cwtex 稱為前階處理程式 (preprocessor)。 cwtex 程式與 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 系統構成一可排版中文之系統，簡稱為 $\text{c}\text{w}\text{T}_\text{E}\text{X}$ 排版系統。

1.2 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 只是一個排版工具



咖啡! 咖啡?

$\text{L}\text{A}\text{T}_\text{E}\text{X}$ 是一個排版程式，而且是公認排版品質甚佳的排版工具。但工具只是工具，有好的工具並不保證你的文稿能排得完美。舉例來說，家庭用的義式咖啡機中，Rancilio Audrey 是有名的頂級機器。但如果使用不當，你可能燒出如名小說家 Agatha Christie 所說的「名為咖啡的可疑液體」。

好排版的先決條件是排版者必須具備排版的專業知識。簡單來說，排版者必須了解什麼樣的版面才能有效率地將作者的想法傳達給讀者。今天，如果你有一部個人電腦，再加上適當的軟體，你就擁有一套桌上排版系統。但不幸的是，大部分的桌上排版系統所排出的文稿，其品質只有一句話可以形容：遭透了！主要的問題不在於硬體，也不在於軟體，而在於排版者的排版知識與品味。底下且舉幾個例子說明常見的排版問題。

- 正文字體使用楷體

楷體其實不適合閱讀，因此報紙或專業刊物中絕不用楷體編排正文。不過，楷體可用於標題或簡短的引文。以 $\text{c}\text{w}\text{T}_\text{E}\text{X}$ 排版時，若選用楷體排版正文，版面上看來符合流行，但讀者可能苦不堪言。

- 行距太小

$\text{L}\text{A}\text{T}_\text{E}\text{X}$ 原設計於排版英文。中文之筆劃較複雜，故中文文稿的行距應比 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 內定值加大一些。以 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 排版中文時，若直接使用內定之英文行距，你是在虐待你的讀者。

- 數學式與上下文之間距

\TeX 對於數學式子與上下文字之間距設有內定值，這是參考專業意見所定出來的。但有些人喜歡加大間距，有些人喜歡縮減間距。除非你有數式排版之專業知識，否則你的調整通常會產生反效果。因此，至少在初學階段，儘可能學習使用 \TeX 之內定值。

- 註解編號應加於標點符號之後

在 \TeX 文稿中，只要使用 `\footnote` 指令即可排版出註解來。但我們必須注意的是，指令應該下在何處。一個常見的錯誤是把註解編號放在中文字之間，例如，「氣象預報¹ 說，高雄市 ...」，正確的位置是，「氣象預報說，¹ 高雄市 ...」。

- 「中文句點如何置於中央？」

這是 cwTeX 討論區經常出現的問題。如果不怕麻煩，我們可以指令控制讓所有的標點符號（包括數學式內之標點符號）居中排版。不過，在作此努力之前，應該先問的是：為何標點符號要居中排版？你的回答可能是：「大家都這麼作。」但是，大家都這麼作不表示這是對的。

如果你真的在乎這個問題，我們建議你作個實驗。取一篇標準的 Word 文稿，其標點符號是居中的。把其中的逗點替換成半型（英文逗點），其後再加一個半型空格。比較看看那一個排版易於閱讀。再進一步，把這篇文章以 \TeX 重排一次，仔細比較看看，標點符號居中到底有沒有道理。

專業的排版知識需長時間才能累積出來，但基本的排版概念並不難學習。有了基本的排版概念就可以避免一些常見的錯誤。本書主要介紹 \TeX 指令，但也介紹一些基本的排版觀念。

1.3 本書章節

本書是 cwTeX 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 cwTeX 系統，請先安裝程式。第 4 章說明如何安裝 cwTeX 系統。初學習排版，請先閱讀 2-6 章。第 2 章提供 6 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，也介紹如何使用 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 章說明文稿結構的概念，第 8 則介紹段落排版之指令。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。第 9 章說明數學式之排版，

對某些人來說，這可能是 $\text{T}_{\text{E}}\text{X}$ 最引人入勝之處。第 10 章介紹表格編排。表格排版指令較複雜，因此我們提供較多的例子說明。

$\text{E}_{\text{T}}\text{X}$ 的發展與 PostScript 繪圖語言有相當密切的結合。此一結合讓我們可以使用各種專業品質之字體、引用外製圖形、或者在文稿中直接繪製圖形。第 11–12 章即介紹圖形與彩色之指令與巨集套件。第 13 章主要說明如何排版投影片與信函。

$\text{E}_{\text{T}}\text{X}$ 系統事實上是以 $\text{T}_{\text{E}}\text{X}$ 為骨幹的一套巨集指令，第 14 章說明巨集指令之概念及定義方法。一般的論文寫作不可避免地須引用參考文獻。 $\text{E}_{\text{T}}\text{X}$ 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_{\text{T}}\text{E}_{\text{X}}` 特別提供 `cw_bibtex` 程式以便能同時排版中英文參考文獻。第 15 章即介紹此一系統。

$\text{T}_{\text{E}}\text{X}$ 系統的能力足以排版任何長度的複雜書籍，第 17–19 章說明書籍排版的細節。排版書籍可能選用特別的字體，因此，第 16 章首先介紹 PostScript 字體之概念及使用方法。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。第 18 章說明如何編製索引。第 19 章以一個實際例子說明如何排版書籍。

網路在日常生活中的角色日趨重要，網路出版的需求也是日趨上升。第 20 章說明網路出版的相關工具。`cw_{\text{T}}\text{E}_{\text{X}}` 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，我們偶而還是需要排版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 21 章說明造新字的方法。`cw_{\text{T}}\text{E}_{\text{X}}` 為幕後排版系統，下指令或輸入文字時，難免出現錯誤。第 22 章說明如何偵測錯誤。最後，第 23 章簡單介紹 `cw_{\text{T}}\text{E}_{\text{X}}` 所提供之字體及程式。

2 例子

學習排版的捷徑是參考現成的例子。本書從第5章開始有許多例子可供參考,但這些例子較短,主要說明特定之排版指令。本章的例子則是說明如何排版一篇文稿。文章寫作須注重結構(structure)。舉例來說,本書共計23章,每一章內有數節,節之下可能有小節;這是本書的結構。一般的論文會分成數節,每一節之下又有小節;這是論文的文稿結構。

文稿應注重結構,因為有清楚的結構才能有效率地傳達作者的想法。對應這個概念, \LaTeX 提供完整的文稿結構指令。本章的例子主要說明 \LaTeX 的文稿結構。如果你把 c\LaTeX 安裝於硬碟 $c:$, 本章的例子檔案全部都置於

```
c:\texmf\cwtex\examples
```

檔案夾內,供讀者自行測試,檔名分別是 `examp1.ctx`, ..., `examp6.ctx`。若你排版這些例子,結果會與本章所呈現的稍有不同,原因是我們在排版本書時,為了易於閱讀,將版面縮小為標準格式(A4紙張)的70%。另外,本書使用較特別的英數字體排版。

除了本章的例子之外,例子檔案夾下還有數個子目錄,存放各種排版例子:

- `\thesis`: 博碩士論文的例子,
- `\book`: 整本書之排版,
- `\beamer`: 單槍投影片,
- `\letter`: 排版信函,
- `\misc`: 申請國科會計畫之檔案等,
- `\html`: 由 \LaTeX 轉換為 HTML 檔案之例子。

c\LaTeX 安裝請見第4章,排版方法請見第3章。安裝完成之後,依第3章之說明,啟動文字編輯軟體開啓例子檔案,即可進行編排。



文稿結構

經濟學是什麼?

吳聰敏

2005.1.4

這是一個老掉牙的問題，每一個人都有自己喜歡的答案。這個講法別人早已說過，我只是用我自己的方式再講一遍而已。

1 矮人頭骨與登山小屋

我的故事從 Flores 矮人開始講起。2003年考古學家 Mike Morwood 及其研究團隊在印尼的 Flores 島上挖到一個小小的人類頭骨。經過分析，這是約 18,000年前存活在島上的人類的頭骨。如果這個頭骨只是一個早期現代人的頭骨，這並不值得那麼興奮。但是，這個頭骨有幾個特別的地方。首先，這不是一個現代人頭骨，Flores 島上的這個人種與目前地球上 60 億人口並無直接關係。

2 從觀察到解釋

Flores 矮人頭骨與登山小屋有何關係? 以內容而言，矮人與山屋毫無關係。不過，Flores 矮人頭骨挖出來之後，科學家開始提問，也嘗試回答問題。各

1

國的登山小屋排在一起看，登山客也開始問問題，也嘗試提供答案。從觀察、提問、到提出解釋，矮人與山屋有類似之處。

各國山屋為何不同? 這是一個經濟學的問題。Flores 矮人頭骨與山屋作對照，表示自然科學與社會科學與並無兩樣，唯一的不同是，自然科學研究自然現象，經濟學是社會科學，研究的是社會現象。

2.1 正確的解釋

經濟學的目的是在解釋社會現象。每一個理論的開頭都是一個猜測，從科學研究的角度來看，無法驗證的猜測是沒有意義的，錯誤的猜測也是沒有意義的。我們要的是正確的解釋，這是科學研究的重點。如果有個科學家告訴你：「我在國際期刊上發表許多論文。」你應該問他：「你的故事是正確的吗?」

3 經濟學有什麼用?

經濟學有什麼用? 如果經濟學的分析是正確的，那麼經濟學的確會有一點用處。

2

```
1 \documentclass[12pt,a4paper]{article}
2 \title{經濟學是什麼?}
3 \author{吳聰敏}
4 \date{2005.1.4}
5 \begin{document}
6 \maketitle
7 \fontsize{12}{20pt}\selectfont
8 這是一個老掉牙的問題，
9 每一個人都有自己喜歡的答案。
10 這個講法別人早已說過，
11 我只是用我自己的方式再講一遍而已。
12
13 \section{矮人頭骨與登山小屋}
14 我的故事從 Flores 矮人開始講起。
15 2003年考古學家 Mike Morwood 及其研究團隊在印尼的
16 Flores 島上挖到一個小小的人類頭骨。
17 經過分析，這是約18,000年前存活在島上的人類的頭骨。
18 如果這個頭骨只是一個早期現代人的頭骨，
19 這並不值得那麼興奮。
20 但是，這個頭骨有幾個特別的地方。
21 首先，這不是一個現代人頭骨，Flores
22 島上的這個人種與目前地球上60億人口並無直接關係。
23
24 \section{從觀察到解釋}
25 Flores 矮人頭骨與登山小屋有何關係?
26 以內容而言，矮人與山屋毫無關係。
27 不過，Flores 矮人頭骨挖出來之後，科學家開始提問，
```

```
28 也嘗試回答問題。
29 各國的登山小屋排在一起看，登山客也開始問問題，
30 也嘗試提供答案。
31 從觀察、提問、到提出解釋，
32 矮人與山屋有類似之處。
33
34 各國山屋為何不同? 這是一個經濟學的問題。
35 Flores 矮人頭骨與山屋作對照，
36 表示自然科學與社會科學與並無兩樣，
37 唯一的不同是，自然科學研究自然現象，
38 經濟學是社會科學，研究的是社會現象。
39
40 \subsection{正確的解釋}
41 經濟學的目的是在解釋社會現象。
42 每一個理論的開頭都是一個猜測，
43 從科學研究的角度來看，
44 無法驗證的猜測是沒有意義的，錯誤的猜測也是沒有意義的。
45 我們要的是正確的解釋，這是科學研究的重點。
46 如果有個科學家告訴你：「我在國際期刊上發表許多論文。」
47 你應該問他：「你的故事是正確的吗?」
48
49 \section{經濟學有什麼用?}
50 經濟學有什麼用?
51 如果經濟學的分析是正確的，
52 那麼經濟學的確會有一點用處。
53 \end{document}
```

例 2.1: 文稿結構

TeX 文稿包含文字與指令兩部分，排版指令皆以反斜線 \ 起頭。例 2.1 的第 1 行指令為：

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

這幾乎是所有 TeX 文稿的第一道指令。顧名思義，\documentclass 選定文稿類別 (document class)，方括號內為指令選項。本例使用 article 文稿類別排版。若是排版書籍，可選用 book 文稿類別。

文稿類別一旦決定，文稿之結構也決定。但是，文稿之版面仍有一些調整的空間。譬如，方括號內之 12pt,a4paper 為指令選項：正文選用 12pt 之字體，紙張為 a4paper。若不加選項，以上指令可簡化成：\documentclass{article}，TeX 將以內定 10pt 字體排版，紙張則選用 letter size。

TeX 提供指令排版題目，作者，及日期，指令分別是 \title, \author, 與 \date (2-4 行)。題目字體會自動放大，且居中排版。如果你不喜歡 TeX 內定之格式，也可以一般指令自行編排。在 TeX 的文稿結構中，正文與題目是獨立的兩部分。正文必須排版於 document 指令環境 (environment) 之內：

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

例 2.1 中，第 2-4 行的標題指令之後，第 5 行為 \begin{document} 表示以下為正文；第 53 行的 \end{document} 則表示正文結束。第 6 行之 \maketitle 表示此行以上為標題。例 2.1 中，\begin{document} 指令位於 \maketitle 指令之前，但兩行指令若對調也可以，對排版結果並無影響。

本例之 \documentclass 指令選擇以 12pt 之字體排版，此一選項同時也選定了行距大小。TeX 之設計是以英文為對象，內定之行距較小，對中文並不適用。因此，排版中文稿時，我們須重新設定行距。第 7 行指令：

```
\fontsize{12}{20pt}\selectfont
```

選定字體為 12pt，行距為 20pt。1pt 等於 1/72.27 吋，故 20pt 約為 0.7 公分。

文稿通常會分段；輸入時空一行 (或多行)，即產生區分段落之效果。本例之第 33 行即以空一行分段。此外，短文通常分為數節，本例計有 3 節，以 \section 指令排版。第 2 節之內還有一小節，其標題是以 \subsection 指令排版；下一個例子將進一步說明章節指令。



中文稿行距

經濟學是什麼？

吳聰敏

2005.1.4

這是一個老掉牙的問題，每一個人都有自己喜歡的答案。這個講法別人早已說過，我只是用我自己的方式再講一遍而已。

1 矮人頭骨與登山小屋

我的故事從 Flores 矮人開始講起。2003年考古學家 Mike Morwood 及其研究團隊在印尼的 Flores 島上挖到一個小小的人類頭骨。經過分析，這是約 18,000年前存活在島上的人類的頭骨。如果這個頭骨只是一個早期現代人的頭骨，這並不值得那麼興奮。但是，這個頭骨有幾個特別的地方。首先，這不是一個現代人頭骨，Flores 島上的這個人種與目前地球上 60 億人口並無直接關係。

2 從觀察到解釋

Flores 矮人頭骨與登山小屋有何關係？以內容而言，矮人與山屋毫無關係。不過，Flores 矮人頭骨挖出來之後，科學家開始提問，也嘗試回答問題。各

1

國的登山小屋排在一起看，登山客也開始問問題，也嘗試提供答案。從觀察、提問、到提出解釋，矮人與山屋有類似之處。

各國山屋為何不同？這是一個經濟學的問題。Flores 矮人頭骨與山屋作對照，表示自然科學與社會科學與並無兩樣，唯一的不同是，自然科學研究自然現象，經濟學是社會科學，研究的是社會現象。

2.1 正確的解釋

經濟學的目的是在解釋社會現象，每一個理論的開頭都是一個猜測。¹我們要的是正確的解釋，這是科學研究的重點。如果有個科學家告訴你：「我在國際期刊上發表許多論文。」你應該問他：「你的故事是正確的嗎？」

3 經濟學有什麼用？

經濟學有什麼用？如果經濟學的分析是正確的，那麼經濟學的確會有用處。

¹從科學研究的角度來看，無法驗證的猜測是沒有意義的，錯誤的猜測也是沒有意義的。

2

```
1 \documentclass[12pt,a4paper]{article}
2 \usepackage[small,sf]{titlesec}
3 \ctxfdef{\section}{\ctxfxr}
4 \ctxfdef{\subsection}{\ctxfxr}
5 \ctxfdef{\title}{\ctxfbb}
6 \title{經濟學是什麼?}
7 \author{吳聰敏}
8 \date{2005.1.4}
9 \begin{document}
10 \maketitle
11 \fontsize{12}{20pt}\selectfont
12 這是一個老掉牙的問題，
13 每一個人都有自己喜歡的答案。
14 這個講法別人早已說過，
15 我只是用我自己的方式再講一遍而已。
16
17 \section{矮人頭骨與登山小屋}
18 我的故事從 Flores 矮人開始講起。
19 2003年考古學家 Mike Morwood 及其研究團隊在印尼的
20 Flores 島上挖到一個小小的人類頭骨。
21 經過分析，這是約18,000年前存活在島上的人類的頭骨。
22 如果這個頭骨只是一個早期現代人的頭骨，
23 這並不值得那麼興奮。
24 但是，這個頭骨有幾個特別的地方。
25 首先，這不是一個現代人頭骨，Flores
26 島上的這個人種與目前地球上60億人口並無直接關係。
27
28
```

```
29 \section{從觀察到解釋}
30 Flores 矮人頭骨與登山小屋有何關係?
31 以內容而言，矮人與山屋毫無關係。
32 不過，Flores 矮人頭骨挖出來之後，科學家開始提問，
33 也嘗試回答問題。
34 各國的登山小屋排在一起看，登山客也開始問問題，
35 也嘗試提供答案。
36 從觀察、提問、到提出解釋，矮人與山屋有類似之處。
37
38 各國山屋為何不同？這是一個經濟學的問題。
39 Flores 矮人頭骨與山屋作對照，
40 表示自然科學與社會科學與並無兩樣，
41 唯一的不同是，自然科學研究自然現象，
42 經濟學是{\ctxfbb 社會科學}，研究的是社會現象。
43
44 \subsection{正確的解釋}
45 經濟學的目的是在解釋社會現象，
46 每一個理論的開頭都是一個猜測。{\footnote{
47 從科學研究的角度來看，
48 無法驗證的猜測是沒有意義的，錯誤的猜測也是沒有意義的。}
49 我們要的是正確的解釋，這是科學研究的重點。
50 如果有個科學家告訴你：「我在國際期刊上發表許多論文。」
51 你應該問他：「你的故事是正確的嗎？」
52
53 \section{經濟學有什麼用?}
54 經濟學有什麼用?
55 如果經濟學的分析是正確的，那麼經濟學的確會有用處。
56 \end{document}
```

例 2.2: 選用字體

本例之文字內容與例 2.1 相同，主要的差別是節標題選用不同之字體。46–48 行則說明排版註解的指令。以 `article` 文稿類別 (document class) 排版時，文稿可區分節與小節等。若是使用 `book` 文稿類別排版，則文稿可能分為數章 (`\chapter`)，每一章之下再分節與小節。節標題是以 `\section` 指令排版，小節標題之指令是 `\subsection`。第 17 行為排版節標題的指令：

```
\section{矮人頭骨與登山小屋}
```

節標題會自動編號，並選用較大的粗黑字體排版；其上下會多留一點空白。

由前面例 2.1 之排版結果可知，節之數字編號是以粗黑字體編排，但中文標題則仍然是明體字，並未變成粗體字。 \TeX 之原始設計是排版英文書稿， $\text{c}\omega\text{T}\text{E}\text{X}$ 擴充其功能以排版中文。不過，因為中英文字體之性質不同，故 \TeX 某些排版指令只對英文有效，選用字體之指令就是一個例子。若覺得節編號數字與中文字體之搭配並不理想，我們須改變數字之字體或者中文之字體。

一篇文稿通常分為數節，改變字體時最好是作全文設定。否則的話，若每一節字體都分別以指令設定，容易出錯。本例之第 2 行，

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

以 `\usepackage` 指令選用 `titlesec` 巨集套件，目的是控制章節標題之排版。巨集套件是一整套排版指令之集合，提供特定之排版功能。 \TeX 之 `\section` 指令能變更英文標題之字體，但設定不易。`titlesec` 巨集套件提供更完整，更簡單之章節標題格式之控制指令。第 2 行指令中之 `small` 選項是讓字體放大級數小一些。此外，`sf` 選項是設定節標題以 `sans serif` 字體排版，這是類似中文圓體之字型。

不過，`sf` 選項則只能變更英數字體，無法改變中文字體，故中文字體須另行以指令設定。本例第 3 行：

```
\ctxfdef{\section}{\ctxfr}
```

設定文稿內 `\section` 指令內之中文都以圓體字排版。同理，第 4–5 行指令分別設定 `\subsection` 與 `\title` 指令內之中文字體。若文稿內某一小段文字要改變為粗黑體，指令如第 42 行所示：

```
經濟學是{\ctxfbb 社會科學}，研究的是社會現象。
```

字體改變之範圍是以左右大括號界定。



標題之字體

```

1 This is cwTeX preprocessor, version 18.3g.
2 **examp2x.ctx
3 (examp2x.ctx [55])
4
5 !Too many {'s or \begin{...}.
6 Output written on c:\xtemp\examp2x.tex & cinput.tex.

```

圖 2.1: cwTeX 程式之訊息: examp2x.xlg

TEX 是所謂的幕後排版系統。排版時，我們先以文字編輯軟體輸入文字與指令，再執行 latex 程式排版。不過，輸入文字與指令時難免發生錯誤，有時候是打字錯誤，有時候是指令下錯了。若是前一種情況，排版仍會出現結果。若是後一種，可能排不出結果來；此時，我們須偵錯並改正。為說明錯誤訊息，我們在例 2.2 之文稿中特別製造 3 個錯誤，並將檔案存為 examp2x.ctx。這 3 個錯誤如下：

- 第 1 行改為: `\documentclass[14pt,a4paper]{article}`,
- 第 17 行改為: `\sction{矮人頭骨與登山小屋}`,
- 第 42 行改為: 經濟學是{\ctxfbb 社會科學，研究的是社會現象。

以文字編輯軟體 (如 WinEdt) 開啓 examp2x.ctx，按 [F9] 功能鍵執行 cwTeX 程式時，DOS 視窗內將出現圖 2.1 之訊息。第 1 行說明 cwTeX 程式是 18.3 版，第 2 行說明文稿檔名為 examp2x.ctx，第 3 行內之 [55] 則表示此一檔案計有 55 行。第 6 行訊息表示，cwTeX 程式處理結果儲存於 c:\xtemp 檔案夾內，檔名為 examp2x.tex 及 cinput.tex。

第 5 行的錯誤訊息說明，此文稿的左大括號或者 `\begin{...}` 指令數目太多。TEX 的指令是以左右大括號或者 `\begin{...} ... \end{...}` 界定範圍。一般而言，大括號是左右成對出現；同理，若文稿內有 `\begin{...}` 指令，則必有對應之 `\end{...}` 指令。cwTeX 轉換 examp2x.ctx 檔案之中文字時，發現左右指令數目不對，故發出上列訊息。錯誤的原因是第 42 行漏了右大括號。

本例之訊息較簡單，若是排版長篇文稿，有時候會出現多處錯誤，一個 DOS 視窗畫面無法容納全部的訊息。為方便檢視訊息，cwTeX 程式將所有之排版訊息儲存於 examp2x.xlg 檔案內，存於 c:\xtemp 檔案夾內。若有必要，可以文字編輯軟體開啓閱讀，以方便偵錯。

cwTeX 程式只是將檔案中之中文轉換為 TEX 格式，實際排版須靠 latex 程式。執行 latex 程式排版時，DOS 視窗上會出現許多訊息，其中絕大部分是關於文稿所使用之中英文之字體。多數訊息都是一閃而過。但如有較重大的錯誤時，

```

1 This is e-TeX, Version 3.141592-2.2 (MiKTeX 2.4) ...
2 entering extended mode
3 **examp2x.tex
4 ...
5
6 LaTeX Warning: Unused global option(s):
7   [14pt].
8 ...
9
10 ! Undefined control sequence.
11 1.28 \sction
12     {\MbQ\cH42}\z{\McQ\cH104}\z{\MaQ\cH244}...
13 ?
14 ...
15
16 ) [1
17
18 ] [2] (examp2x.aux) )
19 ...
20 Output written on examp2x.dvi (2 pages, 4396 bytes).

```

圖 2.2: latex 程式之訊息: examp2x.log

DOS 視窗上之顯示會暫停，以讓排版者有機會閱讀。一般而言，如果按下 [Enter]，排版程式即嘗試自行解決此一錯誤訊息。完整的 latex 排版訊息會記錄於 examp2x.log 檔案內，圖 2.2 只列出部分之訊息內容。

ET_EX 是以 T_EX 為基礎，第 1 行顯示版本號碼，並說明這是 MiK_TE_X 系統。第 3 行說明，latex 程式處理的是 examp2x.tex，這是 cwtex 程式處理 examp2.ctx 之後所產生的檔案。第 6 行出現 LaTeX Warning，下一行則指出 [14pt] 選項不正確。依 ET_EX 之指令設定，\documentclass 僅能選用 10pt, 11pt, 與 12pt 三個選項之一。故當第 1 行之選項改為 14pt 時，排版時即出現錯誤。

第 10 行的訊息為 Undefined control sequence，所謂 control sequence 就是 ET_EX 的排版指令。故此行是說文稿中使用了一個未曾定義 (undefined) 之指令。第 11 行進一步指出，此未定義之指令為 \sction，位置是在文稿的 1.28 (line 28)，也就是第 28 行。錯誤的原因是我們誤把 \section 輸入為 \sction。最後，第 3 個錯誤是「社會科學」四字變更為粗黑體時，忘了加上右括號。此一錯誤不算嚴重，只是 latex 會把「社會科學」開始以下所有的中文字改為粗黑體。

圖 2.2 最後一行之訊息說明，排版結果儲存於 examp2x.dvi 檔案內，共計 2 頁。因為指令有錯誤，故我們須回到原始之 examp2x.ctx 檔案，修正錯誤，重新執行 cwtex 與 latex 排版。一般而言，cwtex 之訊息內容較為簡單，latex 之訊息較為完整。若由前者之訊息內容無法判斷問題所在，應該進一步檢視 latex 之訊息。



文稿類別選項

排版的原則

吳聰敏
2005.03

排版的目的是為了有效率地表達你的意見，讓讀者容易閱讀或吸收。

1 常見的排版錯誤

桌上排版系統普及之後，每個人都能排版，但這不表示每個人都能作出好的排版。關於字體之使用，常見的錯誤如下：

- 楷體與仿宋體等可用於標題，但不宜用於正文；正文請使用明體。
- 較長之英文段落應使用純英文字體編排，如 Times New Roman, Book Antiqua 等。使用中文之細明體或楷體之英文字體，結果保證慘不忍睹。
- 欲強調之字詞，請使用粗黑體或圓體，勿加底線。

Windows 系統僅提供明體與楷體兩種字體，許多人只能選用楷體排版正文或投影片，但排版效果奇差無比。

除了字體之外，版面應注意之細節如下：

1. 文章前端請寫下題目，作者名字，日期。
2. 版面之一行勿拉太長，行距不應過小。
3. 標點符號不應出現在一行前端（避頭點）。

1

4. 標示註解之號碼應在句子標點符號之後。
5. 章節編號使用阿拉伯數字，如 1.1 節, 2.3 節；避免使用「一、一節」或「二、三節」。
6. 阿拉伯數字「1,534」比「一千五百三十四」清楚。

2 圖形與表格

圖形與表格是表達資訊的有效方法，但是圖表作的不好，無法達到目標。常見的表格排版錯誤是格線太多，另一個問題是單位不清楚。大型圖表應移於版面上方或下方，而非排版於段落中間。非萬不得已，表格勿拆為兩頁。

code.ctx

2

```
1 \documentclass[12pt,a4paper]{article}
2 \usepackage[rm,small]{titlesec}
3 \ctxfdef{\section}{\ctxff}
4 \begin{document}
5 \begin{center}
6 {\Large \ctxfbb 排版的原則}\[5mm]
7 吳聰敏\[2mm]
8 2005.03
9 \end{center}
10
11 \bigskip
12 \fontsize{12}{20pt}\selectfont
13 排版的目的是為了有效率地表達你的意見，
14 讓讀者容易閱讀或吸收。
15
16 \section{常見的排版錯誤}
17 桌上排版系統普及之後，每個人都能排版，
18 但這不表示每個人都能作出好的排版。
19 關於字體之使用，常見的錯誤如下：
20 \begin{itemize}
21 \item 楷體與仿宋體等可用於標題，
22 但不宜用於正文；正文請使用明體。
23 \item 較長之英文段落應使用純英文字體編排，
24 如 Times New Roman, Book Antiqua 等。
25 使用中文之細明體或楷體之英文字體，
26 結果保證慘不忍睹。
27 \item 欲強調之字詞，請使用粗黑體或圓體，勿加底線。
```

```
28 \end{itemize}
29 Windows 系統僅提供明體與楷體兩種字體，
30 許多人只能選用楷體排版正文或投影片，
31 但排版效果奇差無比。
32
33 除了字體之外，版面應注意之細節如下：
34 \begin{enumerate}\itemsep=-2pt
35 \item 文章前端請寫下題目，作者名字，日期。
36 \item 版面之一行勿拉太長，行距不應過小。
37 \item 標點符號不應出現在一行前端（避頭點）。
38 \item 標示註解之號碼應在句子標點符號之後。
39 \item 章節編號使用阿拉伯數字，如1.1節，2.3節；
40 避免使用「一、一節」，或「二、三節」。
41 \item 阿拉伯數字「1,534」比「一千五百三十四」清楚。
42 \end{enumerate}
43
44 \section{圖形與表格}
45 圖形與表格是表達資訊的有效方法，
46 但是圖表作的不好，無法達到目標。
47 常見的表格排版錯誤是格線太多，
48 另一個問題是單位不清楚。
49 大型圖表應移於版面上方或下方，而非排版於段落中間。
50 非萬不得已，表格勿拆為兩頁。
51
52 \par\vfill\jobname.ctx
53 \end{document}
```

例 2.3: 排版原則

文稿中常須條列項目，例 2.3 說明條例項目之指令。條列項目可使用 `itemize` 或者 `enumerate` 指令環境。在指令環境內，每一項目都是以 `\item` 指令開頭。兩者不同的地方在於，排版之後，`itemize` 指令環境下每一項目是以一小圓點起頭，而 `enumerate` 指令環境下之項目則是以阿拉伯數字編號起頭。條列指令環境之下可包含另一個條列指令環境。譬如，我們可在例 2.3 的 `itemize` 指令環境下的第一個 `\item` 項目之後（第 22 行之後）加入一個 `enumerate` 指令環境。

排版時，條列項目將自成一個區域，上下多留出一點空白，各項目之間距也比正常行距稍大；圓點與數字編號也會自版面之左沿內縮一點距離。以上這些設定都可以自行調整。例 2.3 的第 34 行：`\begin{enumerate}\itemsep=-2pt`，即設定使項目之間距比內定值小 2pt。

例 2.3 也說明如何自行排版文稿標題。本例之標題包括題目，作者名字，以及日期。這三項都是居中排版，故我們使用 `center` 指令環境（第 5–9 行）。第 6 行排版題目時，選用較大的粗黑體字體：

```
{\Large \ctxfbb 排版的原則}\[5mm]。
```

指令末端之 `\` 為換行指令，指示本行結束；緊接著的 `[5mm]` 則設定行距加大 0.5 公分。若行距為正常行距，則不需 `[5mm]` 之設定。同理，第 7 行排版作者名字之後的 `\[2mm]` 指令，目的也是在加大行距。

本例之第 10 行為空行，空一行表示底下之文字新起一段，故第 11 行開始為新段落。本例第 11 行的 `\bigskip` 指令則設定間距要加大一些。`TeX` 提供幾個現成的加大間距之指令，一個 `\bigskip` 指令之間距等於內定行距之大小。我們也可使用 `\medskip`，或者 `\smallskip`。前者為內定行距的一半，後者為四分之一。

第 52 行的 `\jobname` 指令代表文稿之主檔名。一般而言，`TeX` 檔案是以 `.ctx` 為附加檔案，故 `\jobname.ctx` 可排出完整檔名。上面曾說明，空一行表示新起一段，`\par` 指令也有新起段落之效果（第 52 行）。其後的 `\vfill` 指令之作用則是將段落之間距儘可能拉大。本例中，此一指令下於文稿末端的 `\jobname.ctx` 之前，其效果就是把檔名與上一段文字之間距儘可能拉大，也就是把文稿檔名排於本頁末端。

`TeX` 為幕後排版，故輸入文稿時，每一行文字之右沿不須對齊；但輸入時應考慮的是方便校對與偵錯。仔細看以上幾個例子，可發現文稿之換行都在標點符號或空白處。在 `TeX` 中，換行之效果與空格相同。故輸入文稿時若在兩個中文字之間換行，排版之後，這兩個中文字之間會多出一個空白。因此，輸入文稿時請記住一個重要原則：請在標點符號之後或空格處換行。



輸入原則

台灣長期的經濟發展

吳聰敏

September 26, 2005

以平均每人 GDP 作為經濟發展的指標，則 19 世紀初可以說是全球經濟發展的一個轉捩點。圖 1 比較各國之平均每人所得，縱軸取對數值，因此各線之斜率即代表成長率。

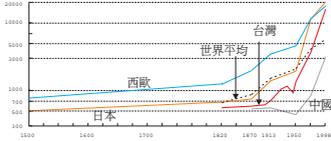


圖 1: 平均每人 GDP

平均預期壽命 (life expectation at birth) 的高低與平均每人 GDP 水準有密切的關係，是經濟發展的一項重要指標。傳統社會之平均預期壽命最低者大約是 25 歲。譬如，英國在 1301--1425 年間平均預期壽命約 24.3 歲，法國在 1740--1749 年間約 24.8 歲。

表 1 比較台灣人、日本人、與中國人之平均預期壽命。1906 年台灣人之平均預期壽命男性僅 27.7 歲，女性為 29.0 歲，這與傳統農業社會之統計

1

表 1: 平均預期壽命

性別	1906	1921	1938	1956
台灣 男	27.7	34.5	41.1	60.2
台灣 女	29.0	38.6	45.7	64.2
中國 男	--	--	--	46.1
中國 女	--	--	--	48.1
日本 男	--	42.1	46.9	62.8
日本 女	--	43.2	49.6	66.8

數字相當。不過，台灣人平均預期壽命在日治時期出現長足的進展，1921 年男性增加為 34.5 歲，1938 年再增加到 41.1 歲。

2

```
1 \documentclass[12pt,a4paper]{article}
2 \usepackage{graphicx,psfrag,booktabs}
3 \renewcommand{\figurename}{圖}
4 \renewcommand{\tablename}{表}
5 \title{台灣長期的經濟發展}
6 \author{吳聰敏}
7 \begin{document}
8 \maketitle
9 \fontsize{12}{18pt}\selectfont
10 以平均每人 GDP 作為經濟發展的指標，
11 則 19 世紀初可以說是全球經濟發展的一個轉捩點。
12 圖 \Z\ref{WorldGDP}\Z 比較各國之平均每人所得，
13 縱軸取對數值，因此各線之斜率即代表成長率。
14
15 \begin{figure}[h]
16 \centering
17 \psfrag{J}{\small 日本}
18 \psfrag{T}{\small 台灣}
19 \psfrag{E}{\small 西歐}
20 \psfrag{C}{\small 中國}
21 \psfrag{W}{\small 世界平均}
22 \includegraphics[width=.8\textwidth]{cx3-GDP.eps}
23 \caption{平均每人 GDP}\label{WorldGDP}
24 \end{figure}
25
26 平均預期壽命 (life expectation at birth)
27 的高低與平均每人 GDP 水準有密切的關係，
28 是經濟發展的一項重要指標。
29 傳統社會之平均預期壽命最低者大約是 25 歲。
```

```
30 譬如，英國在 1301--1425 年間平均預期壽命約 24.3 歲，
31 法國在 1740--1749 年間約 24.8 歲。
32
33 表 \Z\ref{life}\Z 比較台灣人、日本人、
34 與中國人之平均預期壽命。
35 1906 年台灣人之平均預期壽命男性僅 27.7 歲，女性為 29.0 歲；
36 這與傳統農業社會之統計數字相當。
37 不過，台灣人平均預期壽命在日治時期出現長足的進展，
38 1921 年男性增加為 34.5 歲，1938 年再增加到 41.1 歲。
39
40 \begin{table}
41 \small\centering
42 \caption{平均預期壽命}\label{life}
43 \medskip
44 \begin{tabular}{lcrcccc}
45 \toprule
46 & 性別 & 1906 & 1921 & 1938 & 1956 \\
47 \midrule
48 台灣 & 男 & 27.7 & 34.5 & 41.1 & 60.2 \\
49 & 女 & 29.0 & 38.6 & 45.7 & 64.2 \\
50 中國 & 男 & -- & -- & -- & 46.1 \\
51 & 女 & -- & -- & -- & 48.1 \\
52 日本 & 男 & -- & 42.1 & 46.9 & 62.8 \\
53 & 女 & -- & 43.2 & 49.6 & 66.8 \\
54 \bottomrule
55 \end{tabular}
56 \end{table}
57 \end{document}
```

例 2.4: 圖形與表格

文稿中常有機會排版圖形與表格。圖形與表格之排版有許多細節,本例僅簡單說明排版圖形的方法,表格排版詳見第 10 章。L^AT_EX 有內建之畫圖指令,但功能較弱,如果圖形複雜,通常的作法是先由繪圖軟體(如 Corel Draw)製成,轉換成 L^AT_EX 可接受之格式,再引入文稿內。

第22行之 `\includegraphics` 為引入圖形之指令。圖形檔名為 `cx3-GDP.eps`,其中附加檔名 `.eps` 表示此圖檔為 PostScript 格式。這是專業繪圖軟體之標準圖形格式,功能相當強。譬如,圖形可任意放大或縮小,列印品質不會失真。本例中之 `[width=.8\textwidth]` 指令選項,即設定將圖形調整為行長的 80%。

依專業排版之規範,較大之圖表應獨立排於版面的上方或下方,不宜排版於段落中間。為處理圖表之位置,L^AT_EX 提供 `figure` 與 `table` 之浮動圖表(floats)功能。但是,本例之 `figure` 指令環境內因為加上選項 `[h]` (15行),故圖形直接排版於下指令的地方,也就是第一段文字之後。此一排版方式只適用於小圖表。第 40–56 行之 `table` 指令環境是另一個浮動圖表,若依輸入指令之位置,表格的位置應該是在文稿末端。但因為浮動圖表指令之作用,表格排版於頁 2 之上方。

在 `figure` 指令環境內,`\centering` 指令 (16行) 將圖形置於版面橫向之中間; 17–21 行之 `\psfrag` 則是將原圖形中之特定字元替換為另一串字元。例如,第 17 行指令為:`\psfrag{J}{\small 日本}`。排版時,原圖形中之 J 字元將替代以「日本」二字,並以較小之字體排版。事實上,大部分之畫圖軟體都可以直接輸入中文字,因此,本例之 `\psfrag` 指令之作法,目的在求圖形與正文字體一致。另外,如果你畫的圖形須標示複雜的數學式,也可以利用 `\psfrag` 指令來完成,詳見 11.2.2 節(頁 201)之說明。

在浮動圖表指令環境內可使用 `\caption` 指令排版標題,圖表將自動編號。在 `figure` 指令環境內,若為第 2 圖,`\caption` 指令將先排版“Figure 2”,之後再排版標題。本例第 3 行指令將“Figure”一字改為「圖」,故排版結果變為「圖 2」。

文稿內通常以編號徵引圖表,如「參見圖 3」。為方便徵引圖表,我們可在圖表標題之後以 `\label` 指令下標籤,文稿它處再以 `\ref` 指令徵引圖表。本例第 23 行 `\caption` 指令之後加上 `\label{WorldGDP}`,即為下標籤之例子。本例僅含一個圖形,其編號為 1。換言之,WorldGDP 即代表“1”。第 42 行之 `\label` 指令也是同樣用途,但用於表 1。圖表之標籤可以用 `\ref` 指令徵引。第 12 行之「圖\Z\ref{WorldGDP}」指令在排版時將變成「圖 1」。以上指令中加入 `\Z` 指令的目的是在調整「圖」與阿拉伯數字 1 之間距。若不加 `\Z` 指令,間距會小一些,版面效果較差。



圖表位置

台灣匯率制度初探

陳旭昇·吳聰敏*

2005.6

一般探討央行干預如何影響匯率波動的文獻中，多以 GARCH Model 作為實證模型。萬哲鈺 (2000) 曾以 GARCH 模型分析台灣的匯率政策。本文以 Hamilton (1989) 的 Markov switching model 來檢視央行的干預是否能有效地穩定匯率波動。將 Markov switching model 應用於央行對匯率干預的研究，首見於 Beine et al. (2003)。

假設匯率波動為一雙狀態的 Markov switching model，其中 $S_t = H$ 或 L 分別代表高匯率波動狀態 (不穩定狀態) 與匯率低落波動狀態 (穩定狀態)。狀態的轉換矩陣設定為：

$$P(I_{t-1}) = \begin{bmatrix} p_t^{HH}(I_{t-1}) & 1 - p_t^{LL}(I_{t-1}) \\ 1 - p_t^{HH}(I_{t-1}) & p_t^{LL}(I_{t-1}) \end{bmatrix} \quad (1)$$

其中， $p_t^{ij}(I_{t-1}) = P(S_t = j | S_{t-1} = i, I_{t-1})$ ， I_t 為央行干預金額。

式 (1) 之高匯率波動狀態之轉換機率為：

$$p_t^{HH}(I_{t-1}) = \frac{\exp\{\alpha_0 + \alpha_1 I_{t-1}\}}{1 + \exp\{\alpha_0 + \alpha_1 I_{t-1}\}} \quad (2)$$

*通訊: example@ntu.edu.tw.

高匯率波動狀態之轉換機率則為：

$$p_t^{LL}(I_{t-1}) = \frac{\exp\{\beta_0 + \beta_1 I_{t-1}\}}{1 + \exp\{\beta_0 + \beta_1 I_{t-1}\}} \quad (3)$$

如果央行干預匯市有助於穩定匯率波動，亦即 $\partial p^{HH} / \partial I_{t-1} < 0$ ，而且 $\partial p^{LL} / \partial I_{t-1} > 0$ ，則我們預期 $\hat{\alpha}_1 < 0$ ， $\hat{\beta}_1 > 0$ 。

本文之實證結果與 GARCH 模型之結果類似。當外匯市場處於穩定狀態時，央行干預會提高其進入高波動狀態的機率；反之，當外匯市場處於不穩定狀態時，央行干預會提高其持續停留在高波動狀態的機率。

參考文獻

萬哲鈺 (2000)，「中央銀行台北外匯市場干預行為分析」，《台灣經濟學會年會論文集》，109-125。

Beine, Michel, Laurent, Sebastien, and Lecourt, Christelle (2003)，"Official central bank interventions and exchange rate volatility: Evidence from a regime-switching analysis", *European Economic Review*, 47(5), 891-911.

Hamilton, James D. (1989)，"A new approach to the economic analysis of nonstationary time series and the business cycle", *Econometrica*, 57, 357-384.

```
1 \documentclass[12pt,a4paper]{article}
2 \usepackage{amsmath}
3 \usepackage[sort]{natbib}
4 \setlength{\bibsep}{2pt}
5 \renewcommand{\refname}{\ctxfbb 參考文獻}
6 \title{台灣匯率制度初探}
7 \author{陳旭昇·吳聰敏\thanks{
8 通訊: example@ntu.edu.tw.}}
9 \date{2005.6}
10 \begin{document}
11 \maketitle
12 \fontsize{12}{20pt}\selectfont
13 \noindent
14 一般探討央行干預如何影響匯率波動的文獻中，
15 多以 GARCH Model 作為實證模型。
16 \citew{Wan:2000} 曾以 GARCH 模型分析台灣的匯率政策。
17 本文以 \citew{hamilton:89} 的 Markov switching model
18 來檢視央行的干預是否能有效地穩定匯率波動。
19 將 Markov switching model 應用於央行對匯率干預的研究，
20 首見於 \citew{Beine:03}。
21
22 假設匯率波動為一雙狀態的 Markov switching model，
23 其中  $S_t=H$  或  $L$  分別代表高匯率波動狀態
24 (不穩定狀態) 與匯率低落波動狀態 (穩定狀態)。
25 狀態的轉換矩陣設定為：
26 \begin{equation}\label{Markov}
27 P(I_{t-1})=\left[ \begin{array}{cc}
28 p_t^{HH}(I_{t-1}) & 1-p_t^{LL}(I_{t-1}) \\
29 1-p_t^{HH}(I_{t-1}) & p_t^{LL}(I_{t-1}) \end{array} \right]
```

```
30 \end{array} \right]\mbox{。}
31 \end{equation}
32 其中，  $p_t^{ij}(I_{t-1})=P(s_t=j|s_{t-1}=i,I_{t-1})$ ，
33  $I_t$  為央行干預金額。
34
35 式 (\ref{Markov}) 之高匯率波動狀態之轉換機率為：
36 \begin{equation}
37 p_t^{HH}(I_{t-1}) = \frac{\exp\{\alpha_0 + \alpha_1 I_{t-1}\}}
38 \{1 + \exp\{\alpha_0 + \alpha_1 I_{t-1}\}\},
39 \end{equation}
40 高匯率波動狀態之轉換機率則為：
41 \begin{equation}
42 p_t^{LL}(I_{t-1}) = \frac{\exp\{\beta_0 + \beta_1 I_{t-1}\}}
43 \{1 + \exp\{\beta_0 + \beta_1 I_{t-1}\}\}\mbox{。}
44 \end{equation}
45 如果央行干預匯市有助於穩定匯率波動，
46 亦即  $\partial P^{HH} / \partial I_{t-1} < 0$ ，
47 而且  $\partial P^{LL} / \partial I_{t-1} > 0$ ，
48 則我們預期  $\hat{\alpha}_1 < 0$ ， $\hat{\beta}_1 > 0$ 。
49
50 本文之實證結果與 GARCH 模型之結果類似。
51 當外匯市場處於穩定狀態時，
52 央行干預會提高其進入高波動狀態的機率；
53 反之，當外匯市場處於不穩定狀態時，
54 央行干預會提高其持續停留在高波動狀態的機率。
55
56 \bibliographystyle{ter}
57 \bibliography{cx3ttest}
58 \end{document}
```

例 2.5: 數學式

一般公認，數學文稿的排版是 \LaTeX 最強之處。例 2.5 除了說明數學式之排版外，也簡單說明如何使用 `cwbibtex` 排版參考文獻 (bibliography)。參考文獻之排版詳見第 15 章，本例僅簡單說明基本原則。

數學式可分較簡單之隨文數式 (in-text formula) 以及獨立成行之展示數式 (display formula)。依專業排版之規範，數學式內之符號須以數學斜體字體排版，例如 $Y = ax$ 不能排成 $Y=ax$ ；而且各符號之間距與一般文字稍有不同。爲了處理字體及間距問題， \LaTeX 之數學式須特別標示。就隨文數式而言，排版指令爲 `$Y=ax$`。不管你如何輸入數式， \LaTeX 會自行調整各字元之間距，故上式也可以輸入爲 `$Y = a x$`。

數學式常出現特別符號，例如 α , β , π 等。在 \LaTeX 中，數學符號是以指令輸入；以上 3 個符號之指令分別是：`\alpha`, `\beta`，與 `\pi`。此外，數學式中常有上下標，例如下標符號 π_t 之指令爲 `\pi_t`，而上標符號 p^H 之指令爲 `p^H`。若上下標同時出現，如 p_t^α ，指令爲：`p^\alpha_t`，或者，`p_t^\alpha`。兩個連續之上標符號， p^{HH} ，排版指令爲 `p^{HH}`。

欲排版展示數式，我們可使用 `equation` 指令環境，26–31 行是一個稍爲複雜的例子。使用 `equation` 指令環境排版時， \LaTeX 會自動編號，若文稿中須徵引此一數式，我們可以利用 `\ref` 指令。第 26 行之 `\label{Markov}` 爲數式編號之標籤，第 35 行之 `\ref{Markov}` 則徵引此數式。論文在寫作時，經常須修改，數式編號可能會改變。使用徵引指令可保證徵引數式時不會出錯。若某數式不須編號，則 `equation` 指令環境應改用 `\[... \]`。

在學術論文中，確實徵引文獻相當重要。 \LaTeX 提供 `bibtex` 工具以方便處理參考文獻；不過，此一工具僅適用於英文。欲徵引中文參考文獻，應使用 `cwTeX` 所提供之 `cwbibtex` 程式，其方法可簡單說明如下。文稿所使用之參考文獻須先輸入於文獻資料庫 (database)，每一文獻給予一特定名稱。第 57 行之 `cx3test` 即爲文獻資料庫主檔名，全名爲 `cx3test.bib`。文稿內欲徵引某文獻時，指令如第 16 行所示：

```
\citet{wan:2000} 曾以 ...
```

其中，`wan:2000` 爲該特定文獻之名稱。

排版時，經由 `cwbibtex` 程式之處理，該文獻將由資料庫內取出，作者名字排版於文稿內，完整的文獻則以特定之格式排版於文稿末端。本例第 56 行選用 `cwTeX` 所提供之 `ter.bst` 格式排版。

矮人頭骨與登山小屋

經濟學是什麼?

吳聰敏
台大經濟系

2005.08

吳聰敏 台大經濟系 經濟學是什麼?

矮人頭骨與登山小屋

Flores 矮人

Flores 矮人頭骨



吳聰敏 台大經濟系 經濟學是什麼?

矮人頭骨與登山小屋

Flores 矮人

Flores 矮人

- ▶ 大頭的好處 (與壞處)?
- ▶ 壞處: 需要消耗熱量
- ▶ 好處1: 對付敵人
- ▶ 好處2: 吸引異性
- ▶ Flores island: 無大型獵食動物 + 食物少
- ▶ 頭太大, 餓死的機率高

吳聰敏 台大經濟系 經濟學是什麼?

矮人頭骨與登山小屋

Flores 矮人

學生為何要掃廁所?

- ▶ 台北最乾淨的廁所?
- ▶ 台北味道最重 (最不乾淨) 的廁所?
- ▶ 學生為何要掃學校的廁所?
- ▶ 學生為什麼必須掃老師的廁所?

吳聰敏 台大經濟系 經濟學是什麼?

```

1 \documentclass[14pt,dvips]{beamer}
2
3 \usetheme{Malmoe}
4 \usecolortheme{dolphin}
5 \usefonttheme{professionalfonts}
6 \useoutertheme{miniframes} %{\infolines}
7
8 \linespread{1.2}
9
10 \newenvironment{num}
11 {\leftmargini=6mm\leftmarginii=8mm
12 \begin{itemize}}{\end{itemize}}
13
14 \ctxfr
15 \title{經濟學是什麼?}
16 \author{吳聰敏\ 台大經濟系}
17 \date{2005.08}
18 \begin{document}
19
20 \frame{\titlepage}
21
22 \section{矮人頭骨與登山小屋}
23 \subsection{Flores 矮人}
24 \frame{

```

```

25 \frametitle{Flores 矮人頭骨}
26 \includegraphics[width=.8\textwidth]{cx3-Flores.jpg}}
27
28 \frame{
29 \frametitle{Flores 矮人}
30 \begin{num}
31 \item 大頭的好處 (與壞處)? \pause
32 \item 壞處: 需要消耗熱量 \pause
33 \item 好處1: 對付敵人 \pause
34 \item 好處2: 吸引異性 \pause
35 \item Flores island: 無大型獵食動物 $$$ 食物少 \pause
36 \item \alert{頭太大, 餓死的機率高}
37 \end{num}}
38
39 \subsection{有味道的廁所}
40 \frame{
41 \frametitle{學生為何要掃廁所?}
42 \begin{num}
43 \item 台北最乾淨的廁所? \pause
44 \item 台北味道最重 (最不乾淨) 的廁所? \pause
45 \item 學生為何要掃學校的廁所? \pause
46 \item \alert{學生為什麼必須掃老師的廁所?}
47 \end{num}}
48 \end{document}

```

例 2.6: 投影片

TeX 可以排版類似 Power Point 效果之投影片。排版投影片的巨集套件有幾套,例 2.6 介紹的是 beamer 巨集套件。前面幾個例子都是以 article 文稿類別排版,排版投影片則須使用 beamer 文稿類別。文稿第 1 行仍為 `\documentclass` 指令,但使用 beamer 文稿類別。我們可以使用 `pdflatex` 排版;或者以 `latex` 排版,再以 `dvips` 程式輸出 PostScript,最後再使用 `ps2pdf` 輸出為 PDF。本例以後一方式排版,故第一行指令內加入 `dvips` 選項,若使用前一方式,應加入 `pdftex` 選項。事實上,如果不加選項,用任何一種方法仍可排版出投影片來,但 PDF 檔案內不會有超連結 (hypertext links) 之功能。

投影片的字體不能太小,本例選用 14pt 字體排版。若嫌字體不夠大,我們也可以用 17pt 或更大的 20pt。第 3–6 行為 beamer 提供之投影片選項,第 8 行設定行距。前面例 2.1 (頁 22) 曾說明,中文文稿之行距應比內定值大,該例子並以 `\fontsize` 指令選定字體及行距。本例則以 `\linespread{1.2}` 指令將行距加大為內定值的 1.2 倍。

投影片之內容主要是條列重點。前面例 2.3 (頁 28) 曾介紹條列指令環境。因為經常使用條列指令,而且格式略作調整,故第 10–12 行以 `itemize` 指令環境為基礎,重新定義一新的指令環境,名為 `num`。主要的改變經由 `\leftmargini` 之設定值,調整各條列項目排版時之內縮距離。

每一張投影片之文字與指令應置於 `\frame` 指令內,或者 `frame` 指令環境內。本例是使用前一指令,圖 13.1 (頁 228) 則使用 `frame` 指令環境排版每一張投影片。投影片上可用 `\frametitle` 指令排版標題。在 beamer 文稿裡,我們仍可使用 `\section` 與 `\subsection` 指令。排版之後,節與小節標題會排列於投影片上方。演講時,若講到某一張投影片,該節之標題會明顯反白,這有助於講者掌握演講之節奏。若要排版標與小節之目次,可在第 21 行之後加入下列指令:

```
\frame{\normalsize
\tableofcontents}
```

其中, `\normalsize` 指定使用標準字體之大小排版,不須特別放大。

第 31–35 行指令末端有 `\pause` 指令,其功能是創造動態之效果。排版為 PDF 之後,以 Acrobat Reader 播放時,剛進入第 3 頁時,螢幕上只會出現第一條列項,按下 [Enter] 之後,第二條列項才顯示出來。此種動態效果事實是由兩個 PDF 檔案創造出來的,前一張 PDF 上有一條列項,後一張 PDF 上則有兩個條列項。因此,若某張投影片上有 3 個 `\pause` 指令,實際放映時,會依續出現 4 張投影片。

投影片製作完成之後,我們可能要把 PDF 檔案列印出來,此時應把 `\pause`

指令之功能暫時關閉, 否則會印出多張大同小異的投影片。欲關閉 `\pause` 指令之作用, 請在第一行指令內加入 `handout` 選項,

```
\documentclass[14pt,dvips,handout]{beamer}
```

重新排版之後, 可得適合列印之格式。

`beamer` 內建彩色之功能。我們可以把部分文字段落以彩色排版, 這在投影片內特別有用。第 36 行使用 `\alert` 指令,

```
\alert{頭太大, 餓死的機率高}
```

將欲強調之文字以紅色排版。反之, 如果以 `\structure` 指令替代 `\alert`, 則以上之文字將以藍色排版。

3 排版方法

個人電腦上之排版軟體大都是幕前排版, Word 是最有名的例子。L^AT_EX 則是幕後排版軟體。幕前排版軟體使用上很方便。當我們輸入文字時, 軟體立即編排版面。最後的列印結果和螢幕上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; 再以排版程式編排 (typeset); 最後再開啓軟體預覽 (preview) 編排結果。幕後排版之過程間接, 但如果你追求排作品質, 目前幕後排版軟體仍然是唯一的選擇。

T_EX 可以在不同的作業系統上使用, 本書主要介紹 MiK_TE_X 系統, 這是針對 Windows 作業系統所發展出來的。在 Windows 系統上使用 T_EX, 電腦中須安裝 MiK_TE_X 軟體。不過, T_EX 是幕後排版, 故我們還需要預覽/列印軟體。Windows 系統下有好幾套預覽/列印軟體, 本章將介紹 YAP, 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。

除了排版與預覽程式之外, 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, 此一程式功能甚強, 與 T_EX 結合得很好。但這是商業軟體, 必須付費購買。我們另外介紹 Crimson editor, 此軟體使用上沒有 WinEdt 那麼方便, 但功能相當強, 而且可免費使用。各軟體之安裝, 詳見第 4 章。

3.1 檔案管理

在 Windows 系統下, 個人之文稿檔案通常儲存於特定之檔案夾 (或稱為子目錄)。以 Word 為例, 存取檔案時軟體通常是指向「我的文件」, 但實際上, 文稿檔案可能是儲存於硬碟 \Documents and Settings 之下的子目錄內。若使用者將文稿檔案聯結到桌面上, 即可由桌面上點選檔案開啓, 他甚至不須知道檔案置於硬碟何處。簡單來說, 在 Windows 系統下檔案存取之位置是由各軟體所控制。MiK_TE_X 也有類似的功能, 但操作上稍為複雜一些。

舉例來說, L^AT_EX 文稿有時會引用一些由其他軟體所繪製之圖形, 這些圖形檔案必須置於指定之位置, 否則 latex 排版時找不到。如果你的電腦硬碟區分 c: 與

d:, 我們建議將 \TeX 文稿檔案及外製圖形檔案都儲存於 d:\tex 下的子目錄內。例如, 你正在寫一篇關於經濟成長的文章, 檔名為 growth.ctx, 則文稿與圖形檔即可置於 d:\tex\growth 檔案夾內。cw \TeX 安裝程式已設定搜尋 d:\tex 及其下之檔案夾, 故排版時 latex 可找到所需之圖形檔案。

如果以上之設定不符合你的使用習慣, 你也以自行調整設定, 細節請見 4.3.3 節 (頁 52)。反之, 如果你只是偶而使用圖形, 不想作任何設定, 則最簡單的方法是把圖形檔案全部儲存於 c:\xtemp 檔案夾內。這是 cw \TeX 系統內定之工作檔案夾, 排版程式會自動到其中找所需之檔案。

在 Windows 系統下, 我們可以用「我的電腦」或者「檔案總管」程式管理檔案, 第 4 章將介紹 Total commander, 使用上更方便。

3.1.1 DOS 視窗

\TeX 是幕後排版, 其運作與 Word 軟體不同。輸入文稿之後, 我們須啟動 cwtex 與 latex 程式排版。排版時, 螢幕上會出現一長方形視窗, 我們稱之為 DOS 視窗。程式執行若正確無誤, 工作完成之後 DOS 視窗會自動關閉。反之, 若有錯誤, 視窗內之訊息即暫時凍結, 等待進一步處理。

實際排版時, 我們是經由文字編輯軟體啟動排版程式, 並不須自行動手輸入指令。但有時候我們須處理一些較特別的排版需求。舉例來說, 若索引的內容較為複雜, 文字編輯軟體視窗內指令無法處理時, 我們須自行在 DOS 視窗內執行特定之程式。在 WinEdt 視窗上有一圖像 (icon), 名為 Command Prompt, 直接點選, 即可開啓 DOS 視窗。或者, 在 Windows 桌面上依照下列步驟:

開始 > 所有程式 > 附屬應用程式 > 命令提示字元

也可以進入 DOS 視窗。

3.1.2 批次檔

若自行在 DOS 視窗內啟動程式, 可能是執行某一程式, 也可能是執行批次檔。批次檔英文原名是 batch, 檔案是以 .bat 為附加檔名。顧名思義, 這是把一個或多個指令一次處理完畢。舉例來說, 我們可以使用文字編輯軟體輸入一批次檔 cx.bat, 內容僅有一行如下:

```
cwtex %1.ctx
```

並儲存於 c: 之根目錄。將下一節介紹之文稿檔案 test.ctx 複製於 c:\; 再進入 DOS 視窗, 鍵入以下指令:

```
c:\>cx test
```

按下 [Enter] 之後, 電腦即啟動 cx.bat。程式執行時, 批次檔內之 %1 將替代為文稿之主檔名 test, 因此電腦將執行:

```
c:\>cwTeX test.ctx
```

若批次檔有兩行以上, 電腦將依序執行每一行之指令。

3.2 排版之步驟

TeX 是幕後排版, 排版的第一步是以文字編輯軟體輸入文稿與指令。第 4 章將說明 cwTeX 之安裝方法。安裝之後, 硬碟中將新增 c:\texmf 與 c:\localtexmf 兩個檔案夾。在 c:\texmf\cwTeX\examples 檔案夾下有許多測試檔, 其中之一為 test.ctx。以文字編輯軟體開啓 test.ctx, 其內容如下:

```
\documentclass[12pt,a4paper]{article}
\begin{document}
\fontsize{12}{20pt}\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}
```

第一道指令設定文稿類別為 article, 選用 12pt 字體與 a4paper。第 3-5 行為文稿內容與版面控制指令, 輸入於 document 指令環境之內。第 3 行以 \fontsize 指令選用 12pt 之字體, 行距設定為 20pt。

以中文稿而言, 排版須經過 3 個步驟:

- 執行 cwTeX 將文稿內之中文字轉換為 TeX 字體指令;
- 執行 latex 或 pdflatex 進行排版;
- 預覽/列印排版結果。

圖 3.1 說明排版之步驟, 由最右一欄可知, 可供使用之預覽/列印軟體計有三種, 分別是 GSview, YAP, 與 Reader。

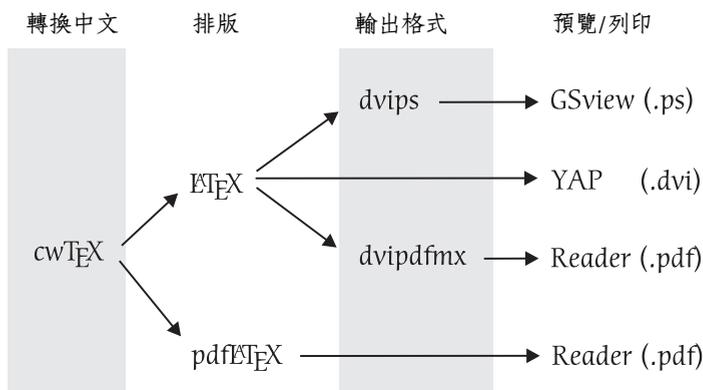


圖 3.1: 排版步驟

若文稿檔名為 `test.ctx`，執行 `cwtex` 程式將文稿內的中文轉換為 $\text{T}_\text{E}\text{X}$ 字體指令之後，結果將儲存為 `test.tex`。接下來以 `latex` 排版，結果將存為 `test.dvi`。如果是用 `GSview` 預覽/列印，它處理的是 PostScript 格式檔案，故 `test.dvi` 須先以 `dvips` 程式轉換成 `test.ps` (見圖 3.1)。PostScript 是 Adobe 公司所創造的繪圖語言，目前是專業排版上的標準。目前網路出版最通用的 PDF 檔案格式，也是從 PostScript 衍生出來的。

我們也可以 `YAP` 程式直接預覽 `test.dvi`。`YAP` 全名為 Yet Another Previewer，這是 `MiKTeX` 內附之預覽軟體。`YAP` 預覽速度快，但如果文稿內引用較特別之圖形，或者排版文字加上變形之效果，`YAP` 可能無法處理，此時應使用 `GSview`。

3.3 文字編輯軟體操作介面

由上一節之說明，輸入 `test.ctx` 文稿之後，我們須執行 3–4 個程式才能預覽/列印排版結果。我們可以在 DOS 視窗內自行鍵入指令啟動程式；但這不方便，也容易出錯。功能較佳的文字編輯軟體都可以由軟體視窗內啟動其他程式。`cwTeX` 系統安裝時，對 `WinEdt` 與 `Crimson editor` 程式之設定作了調整，以方便排版。以下兩小節分別說明之。

3.3.1 WinEdt 操作介面

依照安裝程式所調整之設定，`WinEdt` 視窗上之工作列如圖 3.2 所示。視窗上方靠近中央的位置有一 `cwTeX` 圖像 (icon)。如果目前編輯之檔案為 `test.ctx`，以滑鼠點選 `cwTeX` 圖像，電腦即啟動 `cwtex` 程式，將 `test.ctx` 內之中文轉為 $\text{T}_\text{E}\text{X}$ 字體

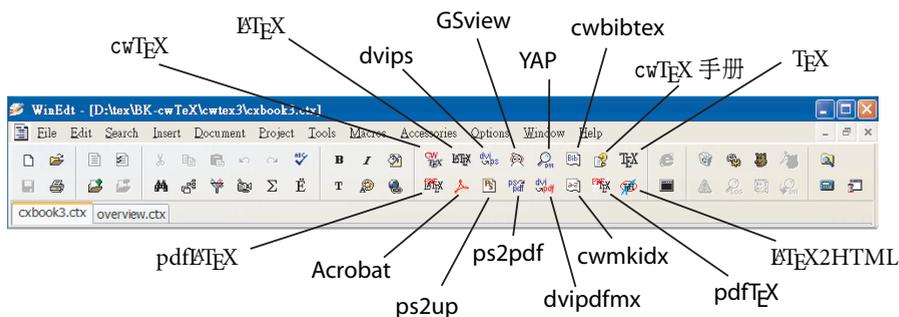


圖 3.2: WinEdt 排版設定

指令,並將轉換結果儲存於 test.tex。如果你習慣使用功能鍵,也可直接按 [F9] 功能鍵,效果相同。

cwTeX 圖像之右方依序是 L^AT_EX, dvips, 與 GSview。因此,我們接著依序點選這三個圖像,排版結果即出現於 GSview 軟體視窗內。如圖 3.3 所示,左邊為 WinEdt 編輯視窗,右邊為 GSview 視窗。在 GSview 視窗上,我們可以將版面放大或縮小;跳到某一頁;也可將排版結果列印出來。如果你回頭修改文稿原檔,則須重新按 cwTeX, L^AT_EX, 與 dvips 圖像。接下來,只須把滑鼠移到 GSview 視窗上,即出現修正後之結果。

以上是以 GSview 預覽/列印排版結果。如果使用 YAP,則如圖 3.1 所示,la-

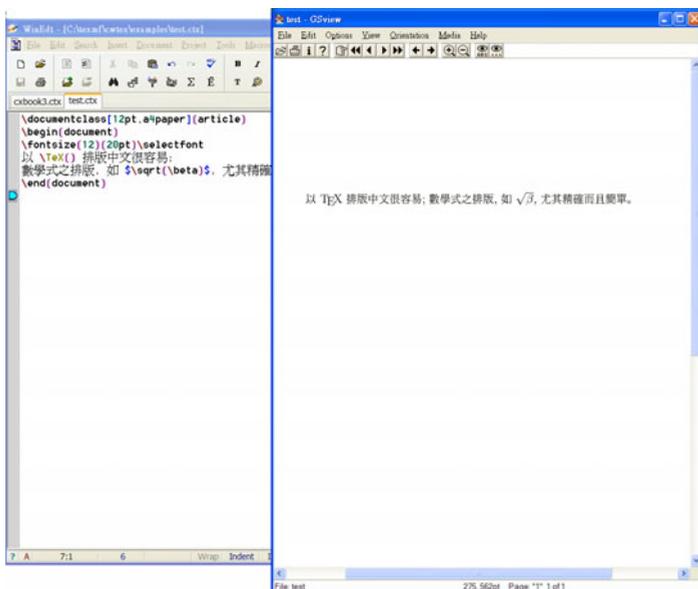


圖 3.3: WinEdt 與 GSview

tex 排版之後, 直接點 YAP 圖像, 即可預覽排版結果。YAP 之預覽速度甚快, 而且 test.dvi 不須先經轉換, 但它無法處理某些圖形效果。

在網路時代, PDF 檔案是文稿交換的標準格式, \LaTeX 排版結果可以經由幾個途徑轉換為 PDF 格式。如圖 3.1 所示, test.dvi 可經由 dvipdfmx 程式轉換為 test.pdf, 再以 Acrobat Reader 預覽/列印。或者, 文稿 test.ctx 先以 cwtex 轉換為 test.tex 之後, 可直接以 pdflatex 排版成 PDF 檔案。pdf \LaTeX 圖像即位於 cw \LaTeX 正下方。

除了以上兩個方法之外, 我們還可以利用 Acrobat Distiller 將 test.ps 轉換為 PDF。Distiller 之功能甚佳, 但這是商業軟體, 須付費購買。最後一個方法是利用 Ghostscript 軟體所提供的 ps2pdf 工具程式作轉換 (位於 GSview 圖像正下方)。以上各個程式都可以轉換出 PDF, 但各程式之功能不盡相同。如果你希望 PDF 檔案可以搜尋中文字詞, 應使用 dvipdfmx 或者 Distiller 作轉換。



中文 PDF

在 dvips 圖像正下方為 ps2up, 其功能是把兩頁的 PostScript 版面縮小並列於一頁上。這一方面可節省列印用紙, 另外這也顯現書籍攤開之後, 兩頁並陳之版面效果。ps2up 是一個批次檔, 內容如下:

```
ren %1.ps %1.psa
psnup -2 -s0.8 -pa4 %1.psa %1.ps
del %1.psa
```

若原檔名為 test.ps, 第 1 行指令先將之改名 (rename) 為 test.psa。第 2 行指令中之 -s0.8 選項指示將每一頁版面縮小為 80%。因此, 從印表機印出之後, 若影印時選擇放大 125%, 結果即為原設計之尺寸。

3.3.2 排版英文稿

如果是純英文的 \LaTeX 文稿, 檔案通常是以 .tex 為附加檔名。因為內容並無中文, 因此不須先以 cwtex 轉換, 可直接以 latex 排版。不過, 如果你是使用 WinEdt 文字編輯軟體, 我們建議的作法是: 開啓檔案之後, 還是依中文稿之步驟排版。雖然第一道步驟 (執行 cwtex) 是多餘的, 但不須特別為英文稿設定另一種排版步驟, 整體而言, 反而是比較省事。



英文稿

3.3.3 Crimson editor 操作界面

WinEdt 是商業軟體, 功能甚佳, 但網路下載之版本僅能試用一個月。期滿之後欲繼續使用, 須付費購買。Crimson editor 則是可免費使用之軟體, 作者為 Ingyu

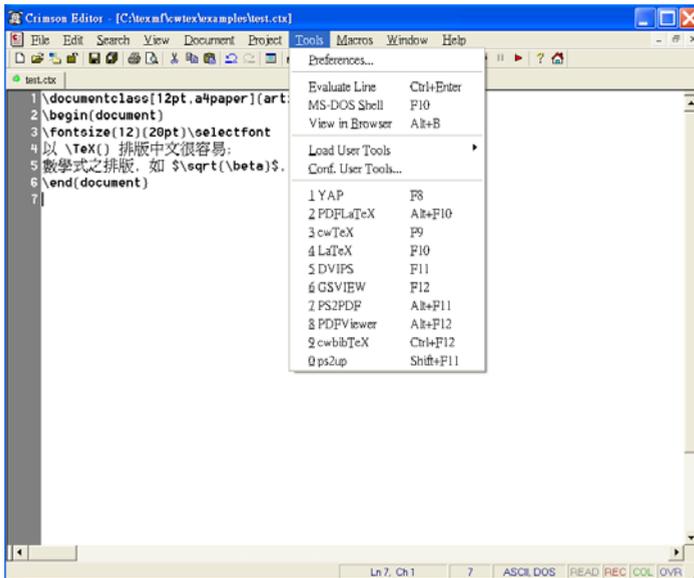


圖 3.4: Crimson editor 排版設定

Kang。此軟體之功能雖然沒有 WinEdt 那麼強，但足供一般人所需。¹

我們也可以在 Crimson editor 視窗內啟動排版程式，但此軟體並無程式圖像之功能，故須透過表單或功能鍵啟動程式。圖 3.4 顯示 Crimson editor 視窗及開啓之 test.ctx 文稿。點選視窗上之 Tools，即出現下拉之表單。其中，第 3 項為 cwTeX 程式，對應之功能鍵為 [F9]。直接點選 cwTeX 選項，電腦即執行 cwTeX 程式，轉換檔案內之中文，另存為 test.tex。我們也可以直接按功能鍵 [F9]，效果相同。接下來，按功能鍵 [F10] 即啟動 latex 排版，按 [F11] 即執行 dvips。最後，按功能鍵 [F12] 即開啓 GSVIEW 軟體預覽結果。啟動其他程式之功能鍵，請見下拉之表單。

3.3.4 工作子目錄

以上兩小節說明，點選 cwTeX 圖像或按下功能鍵 [F9]，即啟動 cwTeX 程式。實際之設定還有一些細節。再以 test.ctx 為例（置於 c:\texmf\cwtex\examples 子目錄下），點選 cwTeX 圖像之後，電腦即執行以下指令：

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

¹Crimson editor 之功能鍵設定曾得到謝子雄先生之協助，我們感謝其幫忙。

表 3.1: 文字編輯軟體之功能鍵設定

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
[F8]	啓動 <code>YAP</code>	[F6]	搜尋/替換字串
Alt+[F11]	啓動 <code>ps2pdf</code>	Ctrl+[F1]	行首加入 % 指令
Shift+[F11]	啓動 <code>ps2up</code>	Ctrl+z	取消上一指令 (undo)
Ctrl+[F12]	啓動 <code>cwibibtex</code>	Ctrl+g	游標移至特定行 (goto)
Alt+[F10]	啓動 <code>pdflatex</code>	Alt+k	刪除游標至行尾之文字
Alt+[F12]	啓動 <code>Reader</code>		

其中, `-d=c:\xtemp` 為 `cwtex` 程式之選項, 指定將轉換之結果 `test.tex` 儲存於 `c:\xtemp` 檔案夾內。如果執行程式時不加入工作子目錄之選項, 執行結果將儲存於 `test.ctx` 同一檔案夾內。

`cwTeX` 系統安裝時即將 `c:\xtemp` 設定為工作子目錄。文稿原檔可以置於硬碟任何地方, 但 `cwtex` 程式會將轉換後之檔案移入工作子目錄內, 後續之排版也全部在工作子目錄內進行。因此, 排版程式所產生之 `test.dvi` 或 `test.ps` 也全部儲存於其中。設定工作子目錄有其方便之處。`TeX` 排版時會產生許多的輔助檔案, 排版完成之後, 這些輔助檔案並無留存之必要。因為所有的輔助檔案都儲存於工作子目錄內, 每隔一段時間可以自行清除掉。

3.4 編輯功能鍵之設定

設定功能鍵的目的是為了方便排版工作。`cwTeX` 安裝程式雖然為 `WinEdt` 與 `Crimson editor` 設定了一些功能鍵, 但使用者也可以自行調整。表 3.1 列出 `WinEdt` 與 `Crimson editor` 之共同設定。左邊兩欄之功能鍵設定, 上一節已經說明; 右邊兩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; [F5] 為搜尋字串; [F6] 則為替換字串。功能鍵 `Ctrl+z` 設定為「取消上一指令」, 這是所謂的 `undo` 功能。譬如, 若你剛剛刪除一段文字, 則按下 `Ctrl+z` 即可回復原文字。

在 `WinEdt` 軟體視窗內下方工作列上有幾個選項, 中間靠右有一項為 [LINE], 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

表 3.2: 輸入中文標點符號之功能鍵

標點符號	功能鍵	標點符號	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+v	〉	Ctrl+Shift+b

方塊區域。

選項 [LINE] 的右邊是 [Spell] 選項, 其右即為 Document Mode 之設定, 原始內定值為 TeX。如果是中文稿件, 我們建議使用 cwTeX 模式。若原先為 TeX 模式, 以右鍵點選, 即出現 Document Settings 視窗, 由視窗右上角之 Document Mode 可選用 cwTeX 模式。以上是以 WinEdt 為例, Crimson editor 也有類似之功能; 請自行研究了解。

3.4.1 輸入中文標點符號

標點符號看起來不顯眼, 但在文稿排版中卻占有舉足輕重的地位。L^AT_EX 提供英文標點符號, 如逗點與驚嘆號, 這些是中英文共用的。但是, 中文有一些特有的標點符號是英文沒有的, 例如中文句點與頓號。

各種中文輸入法都有特定的方法以鍵入中文標點符號。爲了方便起見, 我們在文字編輯軟體內設定鍵入中文標點符號之功能鍵, 如表 3.2。例如, 在文字編輯軟體視窗內按 Ctrl+Shift+o 即出現中文句點; Ctrl+Shift+i 即出現中文頓號。如果要排版兩個中文人名, 例如「阿湯哥·周董」, 人名中間之小圓點可以用 Ctrl+Shift+d 鍵入。

若不作設定, 大部分之中文輸入軟體都是輸入全型標點。但是, 從 L^AT_EX 排版的角度來看, 中英文共用之標點符號, 如逗點或問號, 以半型方式輸入較爲理想。爲方便輸入半型之標點符號, 表 3.2 設定 Ctrl+Shift+k 功能鍵爲半型之逗點; Ctrl+Shift+l 爲半型之問號; 而 Ctrl+Shift+m 則鍵入半型之分號。另外, 欲排版中文數字零「〇」, 請先進入輸入法之全型模式, 再鍵入阿拉伯數字“0”。

4 安裝與設定

cwTeX 系統之安裝包含 MiKTeX 系統, cwtex 程式與中文字型, 文字編輯軟體, 以及預覽與列印軟體。此外, 還有一些有用的工具程式, 如 PostScript 印表機驅動程式等。其中, 文字編輯軟體 WinEdt 為商業軟體, 可試用一個月。預覽軟體 GSview 是 shareware, 但無試用期間之限制。其餘軟體都可免費使用。如果你購買 cwTeX 手冊, 書內所附光碟已收錄所有的軟體與檔案。

如果沒有光碟, 須先下載檔案。網路下載之版本含5套中文字型, 光碟版本則有23套。下載 cwTeX 及相關軟體, 請由 cwTeX 網頁進入:

<http://ccms.ntu.edu.tw/~ntut019/cwtex/cwtex.html>

請注意, 所有檔案合計超過500MB。

4.1 安裝

若電腦內已有舊版之 cwTeX 系統, 請先依 Windows 標準程序移除:

開始 > 控制台 > 新增或移除程式

請先移除 cwTeX, 再移除 MiKTeX。此安裝程式適用於 Win95, Win98, Win2000, WinMe, 以及 WinXP 系統。如果有使用手冊所附之 cwTeX 光碟, 直接執行光碟根目錄內之 cwtex.exe 即開始安裝。cwTeX 光碟含23套中文字型檔。網路下載之版本則含5套中文字型檔。安裝網路下載之版本, 請執行 cwtex-basic.exe。

安裝程式會依續安裝所有的軟體, 並且作一個重要的設定: 調整文字編輯軟體之功能鍵以配合 cwTeX 使用。如果你一開始未安裝文字編輯軟體, 在 cwTeX 系統安裝完成開始運作之後, 仍可再加裝文字編輯軟體, 但須執行一批次檔以調整設定。詳見 4.3 節 (頁 51) 之說明。

安裝時請注意下列細節。

- MiKTeX 系統

MiKTeX 系統包含許多 \TeX 的巨集套件 (package); 一般而言, 我們只會使用到其中的一小部分。MiKTeX 安裝畫面出現之後, 你可選擇 Small, Large, 或者 Total, 三者的主要差別在於安裝多少巨集套件。

MiKTeX 從 2.4 版開始, 已有自動加裝套件之功能。因此, 安裝時應選擇最小套件 Small。未來排版時若須使用其他套件, MiKTeX 可即時自動加裝。選用 Small 選項, 約需 100MB 硬碟空間, 安裝時約需 5–10 分鐘。

其次, 我們須選擇將 MiKTeX 安裝到那一個檔案夾:

請選用內定值!

亦即, 系統安裝於 `c:\texmf`, 而個人設定檔案安裝於 `c:\localtexmf`。萬一電腦之 `c:` 碟空間不足, 你也可安裝於 `d:` 或 `e:`。若選擇硬碟 `d:`, 系統應安裝於 `d:\texmf`, 個人設定檔安裝於 `d:\localtexmf`。

- $\text{c}\omega\text{T}\epsilon\text{X}$ 程式與中文字型

MiKTeX 安裝之後, 接著會安裝 $\text{c}\omega\text{T}\epsilon\text{X}$ 程式與中文字型檔。

- 文字編輯軟體

\TeX 排版須使用文字編輯軟體, 你可選擇安裝 **Crimson editor** 或 **WinEdt**。文字編輯軟體安裝之後, 安裝程式會調整其設定, 以配合 $\text{c}\omega\text{T}\epsilon\text{X}$ 系統之運作。**WinEdt** 文字編輯軟體安裝之後, 會自行啟動。請讓程式啟動之後再將之關閉, 否則安裝程式無法作設定。

- 預覽軟體

安裝 **Ghostscript** 與 **GSview** 時, 安裝程式須把檔案解壓至特定檔案夾內。如果你電腦的「使用者」是取中文名字, 解壓時可能出現錯誤, 因而無法安裝。解決方法如下。在安裝作業完成之後, 由 Windows 之

開始 > 控制台 > 系統 > 進階 > 環境變數

修改視窗上之 **TEMP** 變數, 將其值改為 `c:\xtemp`。按「確定」之後退出, 重新開機。

經過以上設定, 執行光碟 `\util` 檔案夾下之 `gs851w32.exe` (或更新版本) 即可安裝 **Ghostscript**。再執行同一檔案夾下之 `gsv47w32.exe` (或更新版本) 即可安裝 **GSview**。

- 檔案管理工具 Total commander

若文稿引入許多外製圖形，排版程式須能找到所有的檔案。因此，文稿檔案與圖形檔案須適當管理以方便排版作業。安裝 cwTeX 時，你可選擇是否要安裝 Total commander，這是 shareware 軟體，使用方便。設定方法請見 4.3.2 節 (頁 52)。

4.2 測試

欲測試安裝是否成功，請啓動文字編輯軟體 WinEdt (或者 Crimson editor)，開啓 `c:\texmf\cwtex\examples` 子目錄下之 `test.ctx`。依序按下功能鍵 [F9]，[F10]，[F11]。以上功能鍵分別啓動 `cwtex`，`latex`，及 `dvips`。接下來，再按下功能鍵 [F12]，若排版結果出現於 GSview 視窗內，恭喜你，上路了！你可以進一步測試其他的例子。第 2 章介紹幾個例子，原始檔案也置於上述之子目錄內。第 1 個例子檔案即名爲 `examp1.ctx`。

如果你原先安裝 MiKTeX 系統時是選擇 Small，而排版時使用了其他的巨集套件， MiKTeX 會即時自動加裝。如果你是從光碟安裝，因為原始檔案存於光碟上，此時須把光碟插入光碟機內才能加裝巨集套件。如果你有網路連線，也可以設定讓 MiKTeX 從網路下載巨集套件。設定方法是：

開始 > 所有程式 > MiKTeX > MiKTeX Package Manager

由視窗上方之表單上選 Repository，再選擇 Change Package Repository，即可設定或更改巨集套件之資料來源。

萬一測試出現問題，請執行 `c:\texmf\cwtex\util` 檔案夾內之 `testcwi.bat`。此程式會開啓一 DOS 視窗進行測試，若 cwTeX 未正確安裝，測試程式會嘗試診斷問題之所在。測試過程中若出現錯誤，你可以依下列方式將視窗內之錯誤訊息記錄下來。首先，點選 DOS 視窗的左上角，出現下拉視窗。點選「編輯 > 全選」，DOS 視窗內之訊息會反白。再點選視窗的左上角，點選「編輯 > 複製」。最後，開啓文字編輯軟體，按 `Ctrl+v`，即可將 DOS 視窗內之訊息複製於文字編輯軟體內。重覆以上動件，即可依序錄下各畫面之訊息。

4.2.1 常見之安裝問題

若安裝不正確，最常見的症狀是，從文字編輯軟體內啓動 `cwtex` 時，DOS 視窗內出現下列錯誤訊息：

‘`cwTeX`’ 不是內部或外部指令，
可執行的程式或批次檔。

或者下列訊息：

```
Cannot Run:  
cwTeX.exe -d=c:\xtemp test.ctx
```

其中，`test.ctx` 是測試檔案之檔名。

我們先解釋以上訊息的意義。排版的第一步是啟動 `cwTeX` 程式轉換中文字。欲成功啟動排版程式，電腦必須知道 `cwTeX` 程式安裝於何處，辦法是透過所謂的 `path` 環境變數 (environment variable) 之設定。譬如，若 `cwTeX` 程式是安裝於 `c:\texmf\cwTeX` 子目錄內，則 `path` 必須作同樣設定。MiKTeX 以及 `cwTeX` 安裝程式都會依程式之安裝位置設定 `path`；但因為種種原因，有時候就是無法設定正確。如果 `path` 環境變數未正確設定，當文字編輯軟體嘗試啟動 `cwTeX` 程式時，電腦無法找到指定之檔案，因此而發出上述之錯誤訊息。

萬一安裝程式未能正確設定 `path` 環境變數，我們必須手動設定。如果你的作業系統是 Win2000, WinXP, 或更新的系統，可經由「控制台」設定 `path`。反之，如果你使用 Win98 或 Win95，則須在 `autoexec.bat` 檔案內加入 `path` 設定。

- Win2000, WinXP, 或較新之系統

請由

開始 > 控制台 > 系統 > 進階 > 環境變數

點選「系統變數」項下之 `Path` 變數，並選擇「編輯」；在下一個畫面中我們即可修改 `Path` 之設定。若 `cwTeX` 安裝於 `c:`，則該行最前端應該有：

```
c:\texmf\cwTeX;c:\texmf\miktex\bin;...
```

其中，`c:\texmf\miktex\bin` 是 MiKTeX 安裝程式所設定的，`cwTeX` 安裝程式設定的是 `c:\texmf\cwTeX`。若電腦內並無類似之設定，請自行加入；修改之後，請重新開機。

- Win98 或 Win95

在此系統中，硬碟 `c:` 之根目錄下應該有 `autoexec.bat` 批次檔。不過，在某些電腦中，`autoexec.bat` 可能是一隱藏檔。若是如此，請先將檔案屬性改為普通檔案。以文字編輯軟體開啓此檔，其中應有 `path=` 之設定。若目前的設定中並無 `miktex` 與 `cwTeX` 程式路徑之設定，請在檔案末端加入一行指令：

```
path=c:\texmf\miktex\bin;c:\texmf\cwtex;%path%
```

儲存之後，請重新開機。若 `autoexec.bat` 檔案內已有 `miktex` 之設定，則只須加入 `cwtex` 之路徑即可。

除了 `path` 設定問題之外，執行 `cwtex` 時另一個常見的錯誤訊息是：

```
Can't find yap.exe (MikTeX) ...
```

`cwtex` 執行時會先確認電腦內之 `MiKTeX` 能否正確運作；方法是尋找 `MiKTeX` 系統內之檔案 `yap.exe`。若無法找到這個檔案，即發出上述之訊息。如果你已安裝 `MiKTeX`，則出現上述訊息的原因可能是 `MiKTeX` 之 `path` 未設定正確。

4.3 手動設定

`TeX` 排版系統與一般 Windows 軟體不同，它是由文字編輯軟體，排版程式與預覽程式組合而成。每一個軟體可以作適當設定，讓軟體之間能密切配合，排版時會更方便。本節簡單說明各軟體調整設定的方法。

4.3.1 文字編輯軟體

對文字編輯軟體之功能鍵作適當設定，可讓排版工作變得輕鬆。安裝 `cwTeX` 時，安裝程式會試著調整文字編輯軟體之設定，但若你的系統太複雜，可能無法調整成功。若是如此，你仍可自行手動設定。如果你原先未安裝 `WinEdt 5.4`，現欲自行安裝；或者你原先使用的是 `WinEdt 5.3` 或更早之版本，現欲更新為 `5.4` 版。請在安裝軟體之後，執行 `c:\texmf\cwtex\util` 子目錄內之 `setwed54.bat` 以調整其設定。

如果你是使用 `Crimson editor`，也可以自行安裝，並手動調整設定。設定 `Crimson editor` 程式之批次檔名為 `setcme37.bat`，適用於 `Crimson editor 3.7` 版，也是存放於上述之子目錄內。

執行 `cwtex` 及其他排版程式時，可加入一些選項。安裝程式已加入最常用之選項設定值，但你可能偏好自己的設定。欲在文字編輯軟體內新增或改變設定，首先須了解各程式有那些選項。在 `DOS` 內直接鍵入 `cwtex`，畫面上即出現所有的選項。欲在 `WinEdt` 內變更 `cwtex` 之設定值，請參考 23.2 節（頁 350）之說明。

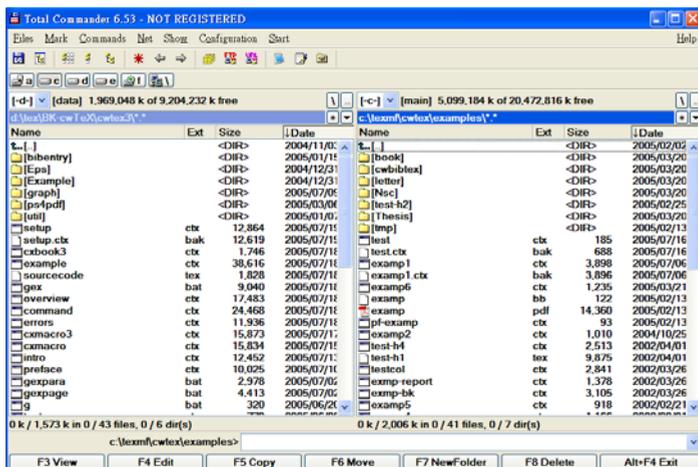


圖 4.1: Total commander 檔案管理程式

4.3.2 Total commander 檔案管理

Total commander 檔案管理軟體之視窗如圖 4.1 所示，左右兩邊之視窗各顯示某一個子目錄內之檔案。視窗內欲顯示那一個檔案夾，可以自行選定。選定之後，要將左視窗內某檔案複製於右視窗之子目錄內，只須以滑鼠選取該檔案，拉至右視窗內即可。

排版的第一步是啟動文字編輯軟體開啓文稿檔案，我們可以在 Total commander 上作設定，以方便開啓文稿。由 Total commander 視窗上

Configuration > Options > Edit/View > Editor

可選用你使用之文字編輯軟體，如 Crimson editor 或者 WinEdt。作以上設定之後，排版的第一步是啟動 Total commander，進入文稿檔案之子目錄，直接點選該檔案，即可啟動文字編輯軟體。

4.3.3 檔案搜尋

排版文稿內可能引用其他圖形檔案。舉例言之，第 2 章例 2.4 (頁 30) 引用了一外製圖形。排版時，latex 程式若找不到該圖形檔，即發出錯誤訊息。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 工作子目錄內。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，最好是設定讓排版程式能搜尋圖形檔。

如果你的硬碟區分 c: 與 d:, c\TeX 安裝程式會設定讓各程式在 d:\text 及其下子目錄搜尋檔案。假設你正在寫作 2 篇論文, 第一篇與經濟衰退有關, 第二篇與流行歌曲有關。則兩篇文稿及附屬圖形檔可分別置於 d:\text\recession 與 d:\text\songs 兩個子目錄內。排版時各程式可自動搜尋, 找到所需的檔案。

如果以上之設定不符合你的習慣, 譬如, 你希望把文稿檔案存放於 e:\paper 之下, 則須調整設定。MiKTeX 系統是依 miktex.ini 之設定搜尋檔案。若系統安裝於硬碟 c:, 此檔案置於 c:\localtexmf\miktex\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, [LaTeX] 標題下有下列設定:

```
Input Dirs=.;\R\text\latex//;\R\text\generic//;\R\text//;\R\cwtex//;d:\text//
```

其中, 等號右邊第一個句點代表檔案所在之子目錄, 接下來的分號區隔下一個設定。%R 代表 c:\texmf 與 c:\localtexmf 兩子目錄。最後的 d:\text// 表示搜尋 d:\text 及其下所有的子目錄。若個人文稿及圖形檔都置於 e:\paper 之下, 請將所有的 d:\text// 改為 e:\paper//。

4.4 軟體更新

c\TeX 系統包括幾個程式及中文字型檔。若程式有新版, 相關檔案可自 c\TeX 下載網址之 \cTeX-update 子目錄內取得。MiKTeX 系統更是常有更新, 其中一部分是 MiKTeX 本身之程式, 另一部分是 \TeX 之套件。相關的資訊會在 MiKTeX 官方網站上宣告。

你可以經由下列方法更新 MiKTeX:

開始 > 所有程式 > MiKTeX > MiKTeX Update Wizard

視窗上會出現 3 個選項, 指定從何處取得更新軟體。如果你的電腦有網路連線, 請選用 Internet 選項。MiKTeX 軟體及 \TeX 套件在許多網址上存有備份, 因此接下來, 你須選擇要從那一個國家的網址下載更新軟體。Taiwan 的網址是中正大學。如果你有寬頻, 也可以選用 UK 三個網址中任何一個。選定之後, 視窗會列出那些軟體可供更新, 若要全部更新, 請按 [下一步]。

要注意的是, 更新 MiKTeX 之後, 可能會出現下列兩個問題:

1. YAP 無法顯示中文,
2. dvipdfmx 無法產生可搜尋中文字詞之 PDF 檔案。



更新 MiKTeX

為解決以上問題，在 MiKTeX 更新之後，請執行 `c:\texmf\cwtex\util` 子目錄下之 `setcwtex.bat`。

4.4.1 加裝字型

如果你安裝的是網路下載的 `cwTeX`，其中僅含 5 套中文字型。若你有舊版 `cwTeX` (修訂 2 版, 2002 年 4 月) 光碟，其中有較多的中文字型。安裝新版之後，你可考慮加裝修訂 2 版光碟上的其他中文字型。不過，`cwTeX` 之中文 Type 1 字型檔曾在 2005 年作了更新。新版中文字型的重要功能是可經由 `dvipdfmx` 或 `Adobe Distiller` 製作可搜尋中文字詞之 PDF；但是舊版上之中文字型檔則無此功能。

你仍然可以把修訂 2 版之中文字型檔加裝至 `cwTeX` 系統內，但是以 `dvipdfmx` 或者 `Distiller` 所產生之 PDF 檔，若有部分文字是以舊版字型排版，這部分之文字無法作中文字詞搜尋。要加裝 `cwTeX` 修訂 2 版光碟上之中文字型，請先確認 `cwTeX` 系統已能運作，將光碟放入光碟機內，執行 `c:\texmf\cwtex\util` 檔案夾下之 `addfont3.bat`。

4.4.2 安裝其他軟體

除了以上軟體之外，還有一些有用的排版工具，可自行加裝。譬如，你可以使用 `ETEX2HTML` 工具程式將 `ETEX` 文稿轉換成 HTML 網頁。或者，如果你常用 `cw-bibtex` 排版參考文獻，你可以安裝 `Jabref` 軟體以管理參考文獻資料庫。以上各程式之安裝將於相關章節內再作說明。

4.5 Linux 版本

`cwTeX` 修訂 2 版中，曾較詳細地說明 Linux 版本之安裝方法。本次之修訂主要是更新 Windows 版本之程式，為了節省篇幅，Linux 程式之安裝說明直接置於網頁上。有意使用者，請進入 `cwTeX` 之 ftp 網址，再進入 `\cwTeX_linux` 子目錄，參閱其中之 `Readme`。此外，請參考李果正先生之網頁：

<http://cle.linux.org.tw/~edt1023/>

或直接至下列網址下載：

<http://edt1023.sayya.org/tex/cw/cwtex-1.63g-linux-1.tar.bz2>

5 指令與文字

\TeX 是幕後排版系統，排版指令與文字必須同時輸入於檔案內。使用 \LaTeX 排版，我們必須了解指令之功能。從本章開始，我們將陸續介紹基本的排版指令。 \LaTeX 的指令完整，可排版各式各樣的文稿。以本書的篇幅而言，我們不可能介紹所有的指令。本書所介紹的指令是以讓使用者能排版出一般的學術論文為目標，我們將特別說明中文排版之相關問題。

排版時，如果遇到問題怎麼辦？首先，我們可以從 \TeX FAQ (常見問題) 找答案，這可由 \TeX 官方網站進入。Google 有一個 \TeX 討論區，常有專家熱心提供答案。若是中文 \TeX 排版的問題，可以試在 $\text{cw}\TeX$ 討論區提問。以上的討論區都可以由 $\text{cw}\TeX$ 網站連結進入。 $\text{cw}\TeX$ 討論區常有人提問，本書各章針對常見問題提出說明，請見第372頁之索引。

如果你的問題與某巨集套件有關，可參考其說明檔。 \LaTeX 所有的說明當都置於 `c:\texmf\doc` 檔案夾下。如果你有興趣深究 \LaTeX 排版的問題，可參考 [Mittelbach and Goossens \(2004\)](#)，或者 [Grätzer \(2000\)](#)；後者主要討論數學排版。



常見問題

5.1 符號與指令

\LaTeX 文稿包含文字與指令。中英文或數字都是直接鍵入。但是，有些符號在鍵盤上並沒有對應的字鍵，譬如，商標符號 © 或英鎊符號 £。有些中文輸入法提供輸入特別符號的方法，因此，在文字編輯軟體的視窗上可看到這些符號。不過，排版時卻會出現錯誤訊息。

在 \LaTeX 中，所有的特殊符號都是以指令排版。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`。 \LaTeX 的排版指令絕大部分是以反斜線開頭，英鎊符號為 `\pounds`，就是一個例子。數學符號也是以指令排版，譬如 π 之指令為 `\pi`，排版 β 之指令為 `\beta`。

\LaTeX 指令是由英文字母組成，而且大小寫是有分別的。例如，`\omega` 可排出 ω ，但若輸入 `\Omega`，結果將變成 Ω 。另一個例子是 \LaTeX 之標識，排版指令是



特殊符號

表 5.1: 重音符號

ò \‘{o}	ö \~{o}	ö \v{o}	ø \c{o}
ó \’{o}	ō \={o}	ő \H{o}	ø \d{o}
ô \^ {o}	ó \. {o}	ōo \t{oo}	ø \b{o}
ö \" {o}	ō \u{o}		

\LaTeX{}。如果大小寫弄錯了，譬如輸入成 \LaTeX{}，將會出現錯誤。你或許注意到 \LaTeX{} 指令之末端加上一對大括號。爲什麼呢？ \TeX 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。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爲止，就構成指令。非字母字元包括空格、標點符號、數字等。因此，若輸入 the \LaTeX logo，排版結果爲：the \TeX logo。 \TeX 標識與其後的文字之間未隔開，原因是指令後面的空格被當成是指令的結束，因此 logo 即緊接著編排。爲了避免此種錯誤，我們應輸入 the \LaTeX{} logo，連續的左右大括號表示指令結束，緊接其後的空格才會排出空白。另一種輸入方法是 the {\LaTeX} logo，右大括號表示指令結束。第三種輸入指令的方法是 the \LaTeX\ logo，其中反斜線加空格是 \TeX 排版空格的指令。

歐洲各國的文字中有一些字母上會加上重音符號 (accents)。表 5.1 列出重音符號 (accents) 及其指令。排版 ö 符號之指令爲 \~{o}；大括號是指令之範圍。若大括號中只有一個字母，則括號可以省略。因此，\~{o} 與 \~o 之效果相同。表 5.1 的例子是把重音加在小寫字母 o 上。實際上，重音符號可以加在任何字母或符號上面。例如，\v{z} 指令可得到 ž，而 \={A}，得到 Ā。

除了重音符號之外，其它常見的特殊符號列於表 5.2。利用這些指令，我們可以排版德文：

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

但若要排版長篇德文稿，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：

表 5.2: 特殊符號

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

今天是 September 26, 2005.

今天是 \today.

若日期要以中文格式排出, 請見圖 13.1 (頁233) 之例子。

有時候我們須在文稿某處排出文稿檔名, 這可使用 `\jobname` 指令。此一指令僅記錄主檔名, 附加檔名須自行填入。譬如, 若文稿檔名為 `code.ctx`, 我們可在文稿末端加入以下之指令以排版檔名與日期:

`code.ctx (September 26, 2005)`

`\jobname.ctx (\today)`

以上指令中, `\jobname` 排版出 `code`, 加上其後的 `.ctx` 才是完整檔名。

5.1.1 註銷指令

輸入文稿時, 有時候我們須在某處加註說明。這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。譬如, 上例中若輸入:

`\jobname.ctx %(\today)`

排版時 `%` 符號本身及其後的文字與指令將完全略過不處理, 故日期及圓括號將不會排出。L^AT_EX 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, 我們馬上碰到問題是: 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 和其他國家相差不大。

台灣政府支出占 GDP 比率約 27%, 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	功能	排版符號之指令
%	註銷	\%
\	定義指令	\$\$backslash\$
{	界定指令範圍	\$\${\$
}	界定指令範圍	\$\$}\$
&	表格排列指令	\&
\$	數式模式	\\$
#	記錄巨集指令參數	\#
~	加入空白或防止分割字串	\~{}
^	數式上標	\^{}
_	數式下標	_{}

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

因為數目字 27 之後緊接著註銷指令，排版時其後的逗號即略過不處理，下一行的文字則緊接在本行末端。要排版百分比 % 符號，我們必須輸入 \%。因此，正確的指令如下：

台灣政府支出占 GDP 比率約 27%，
和其他國家相差不大。

台灣政府支出占 GDP 比率約 27\%，
和其他國家相差不大。

除了註銷指令外，另外還有 9 個符號在 \TeX 中有特殊用途。全部列於表 5.3，以供參考。如表中右欄所示，這些符號本身必須以指令才能排出。其中，註銷指令 % 上面已作過說明，反斜線 \ 是 \TeX 指令之起頭。欲排版反斜線，我們須鍵入 \$\$backslash\$，其中，左右之 \$ 與數式有關，第 9 章將有進一步說明。此外，左右大括號是用來界定指令的範圍。譬如，上面所介紹重音符號時，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。

5.1.2 排版網址

電子郵件地址與網頁網址常含有特殊符號，包括註銷符號與右斜線。要在文稿內排版電子郵件地址與網址，可使用 Donald Arseneau 所寫的 url 巨集套件。使用巨集套件，首先須在 \begin{document} 指令之前的全文設定區引入；之後即可使用其所提供之指令。

```
http://ccms.ntu.edu.tw/~ntu
example@ntu.edu.tw
```

```
\usepackage{url}
\def\UrlFont{\rm}
...
\url+http://ccms.ntu.edu.tw/~ntu+ \
\url{example@ntu.edu.tw}
```

本例中，我們還在全文設定區加入 `\def\UrlFont{\rm}` 指令，目的是設定以標準的 Roman 字體排版網址。

排版網址或電子郵件地址之指令為 `\url`，所要排版之資訊可置於大括號內，或者以特定之符號標示其範圍。本例之第 1 行是以 `+` 標示範圍，第 2 行則以大括號界定範圍。

5.2 英文稿件輸入原則

\LaTeX 是幕後排版，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 \LaTeX 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的，這是不必要的。但是，輸入仍須注意一些細節，否則排版結果可能出乎你的意料之外。本節與下一節將說明英文與中文文稿之輸入原則。

輸入英文稿時要注意一些基本原則。首先，英文單字之間必須留有空白。使用 Word 軟體時，若兩個單字之間鍵入兩個空格，排版之後這兩個單字之間距就會加大。但是在 \LaTeX 文稿中，多個空格與一個空格的作用是相同的，排版後都是一個空白。

另外，換行與一個空格的作用相同。排版之後，本行行末之字元與下一行行首之字元之間會出現一個空白。圖 5.1 的例子中，第 1 行行末為 `are`，第 2 行開頭為 `marked`。表面上看來，兩者之間並未留有空格，但因為 `are` 之後換行，在 \LaTeX 中其效果與留空格的作用相同。排版之後，這兩個單字之間會有正確的間隔。圖 5.1 第 2 行的 `by` 與 `spaces` 之間留有三個空白，排版之後其間距和正常間距相同。這表示數個空格之作用與一個空格相同。

為了方便文稿校對與修改，輸入文稿時一行不宜太長。問題是，應該在那裡換行？根據以上之排版原則可知，輸入英文文稿時應在單字之間，或者標點符號之後換行。標點符號之後本來就須留出空格，故在此處換行不致影響排版結果。

圖 5.1 的兩段輸入文字之間有一空行，其作用是以下新起一段。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。和空格的情況類似，空一百行和空一行的作用完全一樣，都是表示



換行與空格

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

One or more blank lines denote the end of a paragraph. So it is OK to add more blank lines after a paragraph.

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

One or more blank lines denote
the end of a paragraph.
So it is OK to add more blank lines
after a paragraph.
```

圖 5.1: 文稿結構

要另起新的段落。新段落開始的第一行，行首會自動內縮一點。英文排版中，這稱為 indent。如果不要內縮，可在段落之前加上 `\noindent` 指令。

爲了提高可讀性， \LaTeX 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。舉例來說，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. 單字的小圓點並不是英文句點，但 \LaTeX 誤把它當作句點處理，因此 Mr. 與 know-all 之間的空白拉大了一些。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格。因此，正確的輸入方法是第 2 行所示：`Mr.\ know-all`。一般人可能認爲這點小差異沒有那麼了不得。不過，在專業排版者的眼中這個差別是重要的。

此外，英文句點之後若緊接著右圓括號或者英文引號， \LaTeX 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：

Many famous coffee beans (Santos, etc.) are expensive.

```
Many famous coffee beans
(Santos, etc.)\ are expensive.
```

輸入時，右圓括號之後加上強制空白指令，以免間距太大。

5.3 中文稿件輸入原則

輸入英文稿時，單字之間很自然會留下空格；換行也是在單字之間或是標點符號

之後。中文文稿的輸入方法稍有不同，主要是中文字之間不會留出空格。如果不小心在兩個中文字之間留下一個空格，排版之後這兩個中文字之間會出現一個多餘的空白。

一般在輸入中文時，並不會特意在文字之間留下空格，因此這不會出現問題。比較常見的問題是，因為某一行文字實在太長，故輸入時選在文字中間換行。但是，上一節已說明，換行與空格的排版效果相同。意即，若在一行文字當中換行，排版之後換行處會出現一個多餘的空白。底下是一個例子：

輸入中文文稿時，要特別注意空白及換行的位置。

輸入中文文稿時，要特別注意空白及換行的位置。

因為「文稿」兩個字之間有一空格，排版之後兩字之間距不正確。此外，第一行在「注意」之後換行，故排版之後，這兩字之後也出現一多餘的空白。

那麼，中文稿應該在那裡換行呢？排版時，標點符號之後本來就要留一空白。因此，輸入中文稿時我們可以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：



中文輸入原則

- 句子內的中文字之間請勿留下空白；
- 換行請盡量在標點符號之後。

根據這兩項原則，上面句子中應該在「中文文稿時，」的逗點之後換行。

不過，如果你的中文輸入已養成習慣，在固定長度就會按下 [Enter] 鍵換行，則另一個解決問題的方法是在執行 `cwtex` 程式時加入 `-c` 選項。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` 程式會將換行指令刪除，故排版結果為正確。`ctWTeX` 安裝程式已在文字編輯軟體上作以上之設定，但若有特別需求，可自行刪掉此設定。以 `WinEdt` 為例，開啓 `WinEdt` 視窗上方之：

Option > Menu Setup > &Accessories > `cwtex`

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

有時候，我們會碰到非常長的英文單字或句字，若要等到標點之後再換行，輸入時並不方便。此時，我們可以使用註銷指令 `%`。如前所述，註銷指令本身及其後所有的文字符號都不處理。譬如，

很長的單字，例如 percholoethylene，
可以用註銷指令切成兩行；此一方法
也適用於中文。

很長的單字，例如 percho%
loethylene，可以用註銷指令切成%
兩行；此一方法也適用於中文。

此例之句中有一較長的英文單字，輸入時因為一行容納不下，故在單字當中加上 % 指令，這等於把第 2 行開頭的 loethylene 直接接上第 1 行尾巴的 percho，排版結果是正確的。同理，「切成」兩字之後加上 % 指令，因此換行指令不會使排版結果多一空白。

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，也使用於中文內。但中文另外有 11 個標點符號是英文沒有的，例如中文句點與頓號。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，我們在文字編輯軟體內設定鍵入中文標點符號之功能鍵，請見表 3.2 (頁 45)。譬如說，要鍵入中文句點，請按 Ctrl+Shift+o ，而輸入中文頓號之按鍵為 Ctrl+Shift+i 。

目前各種輸入方法中，大都採「全型輸入」方式輸入標點符號。譬如，中文句點在螢幕上占用相當於兩個英文字母的位置。中文特有之標點符號，如中文句點，一定是全型。相對的，中英文共用之逗點或者問號，可能採英文輸入（半型），也可能採中文全型輸入。若是輸入純英文稿，所有文字與標點都是半型，而且我們會在標點符號之後留下一空格。 latex 排版時，標點符號後面留出的空格有一重要的功能，此一空格讓排版程式能調整一行當中之字距。

但中文稿之標點若採全型輸入，習慣上我們不會在標點之後再留一個空格。雖然在顯示器上看來該標點之後有一空格，但其實不然。如果不留空格， TEX 排版時中文標點符號之後也不會留出空白，結果整個版面將擠成一團。因此，輸入 TEX 文稿時，除了中文特有之標點之外，其它全部以英文半型標點輸入，其後並留下一空格才是正確的。不過，你的文稿之標點符號若採全型輸入，而且其後未留下空格，排版時標點符號之後仍有適當間距出現，原因是 cwtex 程式會辨識常見的錯誤，並作修正。雖然 cwTeX 有此功能，使用者最好還是自己養成正確的輸入習慣。

\LaTeX 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, c\LaTeX 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, 不應加上任何空格。如果插入空格, 排版結果反而不理想。

5.3.2 英文引號與破折號

英文標點符號用於中文稿中另外要注意的是引號。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, 如果你要排版底下句子:

“‘Five’ or ‘Hive?’” she asked.

“\, ‘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。

英文使用的破折號由短到長共有三種, 全部列於表 5.4。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: --。若你習慣使用 ~ 符號連接數目字, 例如「1945 ~ 1950 年」, 可使用 \LaTeX 指令排版連接符號。

請注意, 如果使用全型輸入法, 我們可以直接在文字編輯軟體內輸入 ~, 但執行 c\LaTeX 時將出現錯誤訊息。換言之, c\LaTeX 之中文字型檔並無 ~, 但 \LaTeX 的字型檔中有此一符號, 故我們使用 \LaTeX 的指令排版此連接符號。

表 5.4: 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 (−)	$\$-\$$	$5 - 2 = 3$

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:「電力公司即將限電 — 如果天氣不轉涼的話」。但是, 因為中文與英文字母之高低不同, 故以上短線之位置稍偏低。若你不怕麻煩, 指令可調整如下:

電力公司即將限電 $\text{\raisebox{2pt}{---}}$ 如果天氣不轉涼的話

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 $\$$ 的作用是進入數學排版模式。譬如, 輸入 $\$-\100 , 排版結果為: -100 , 若不進入數學模式, 排版結果為 -100 。

有些人以連續三小點 (甚或六小點) 代表一段省略的話。在 \TeX 中, 你可以用 \dots 指令產生三小點。例如指令 \dots\dots 產生:。最後, 謹將輸入標點符號之原則歸納如下。

- 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 cwtex 程式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 \(1996\)](#)。

5.3.3 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。

有些人甚至認為在橫排的版面中，文字上下對齊才好看。結果在排版之後，每一行的中文字上下對得整整齐齊的，但卻無法避頭點。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。遇有這種情況，請將前一兩行的文字增減一字，以解決避頭點的問題。

5.4 排版其他西方國家文字

除了英文之外，TeX 也可以排版德文、法文等歐洲國家文字。TeX 有幾套專門用來排版歐洲國家文字之巨集套件。譬如，`german.sty` 可用以排版德文。不過，由這些巨集套件所衍生的 `babel` 巨集套件使用上更方便，作者是 Johannes Braams。

`babel` 巨集套件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，或參考 [Mittelbach and Goossens \(2004\)](#)。

6 選用字體

開始排版文稿之前，應先想好版面之整體設計，如版面尺寸、正文字體、章節標題格式、註解的字級 (font size) 等等。L^AT_EX 有現成的版面格式可供選用。舉例來說，若以 article 文稿類別排版，且選用 12pt 選項，則正文將以 12pt 字體排版，註解之字體會自動縮小於 10pt。文稿內若區分節與小節，其標題會自動選用較大的粗黑字體。

L^AT_EX 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。使用現成的格式可避免一些常見的排版錯誤，這是其優點。但是，也有人認為 L^AT_EX 所排版的書籍其版面格式都很類似，不易顯出個人的設計風格。事實上，透過現成之巨集套件，要變更版面設計並不難。當然，要自行設計版面，你必須有一些版面設計的基本概念，也必須熟悉相關的巨集指令。字體是排版的最基本元素，本章主要介紹選用中英文字體之指令。

6.1 字體規格

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 T_EX 排版系統時，同時又發展一套設計字體的軟體，稱為 Metafont。原始 L^AT_EX 系統所使用之英文字體全部都是利用 Metafont 軟體設計出來的。這些字體之檔名大部分是以 cm 開頭，代表 Computer Modern。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。1980 年代中期以來，字體設計之技術有相當大的進步，其中又以美國 Adobe 公司所發展之 Type 1 字體 (又稱為 PostScript 字體) 最為重要。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，

表 6.1: 英文字體

	標準體	斜體	粗黑體	斜粗黑體	數學斜體	SMALL CAPITAL
Windows	font 123	<i>font 123</i>	font 123	<i>font 123</i>		
TeX	font 123	<i>font 123</i>	font 123	<i>font 123</i>	<i>font 123</i>	FONT 123

幾乎變成是電腦排版的標準。因為如此, TeX 一個重要的發展方向就是讓排版者可以使用 Type 1 字體。

在 Windows 系統普及之後, 中文電腦字體之技術也有很大的改變。Windows 主要使用另外一種字體規格, 稱為 TrueType, 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。

6.2 選用字體

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 選擇字體之指令只適用於英文與數字, 對於中文字體並無作用。中文字體之變更必須經由 cTeX 的中文字體指令。

6.2.1 英文與數字字體

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。在舊版的 TeX 系統中, 字體指令之彈性較小。1990 年代初期, 兩位德國專家 Frank Mittelbach 與 Rainer Schöpf 設計出「新式字體選用法」(new font selection scheme, 簡稱爲 NFSS)。此套方法出現之後, 頗受歡迎, 現在已經變成新版 TeX 系統之一部分。

在 Word 或其他文書軟體中, 大部分之中英文字體有 3 種變形: 斜體、粗黑體、與斜粗黑體。表 6.1 第 1 列爲一般的 Windows 字體; 第 2 列則爲 TeX 之字體。兩者的主要差別是, TeX 另有數學斜體及 SMALL CAPITAL。數學斜體用於排版數學式, 所謂 SMALL CAPITAL, 是把小寫英文字母以大寫字母的字形寫出, 但字體較小一些。譬如, Taiwan 若以 SMALL CAPITAL 排版, 結果爲 TAIWAN。

表 6.2: 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>

TEX 的字體分類更細，故選用字體之指令也較多。依照 NFSS 之分類，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)。

字形又進一步區分為：直立形 (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 (**bold series**)。以上之分類對應表 6.1 第 2 列之各種字體。

Windows 系統內附許多英文字體。譬如，由 Word 視窗的字體下拉表單中，可看到許多的字體，如 Times New Roman, Arial 等。每一種字體有其設計風格，但大體上仍可歸於幾個大類。在 TEX 中，林林總總的字體可區分為三大類：羅馬字族 (Roman family), sans serif (無裝飾邊) 與打字機 (typewriter) 字族。上述之 Times New Roman 屬於羅馬字族，Arial 屬於 sans serif; 而打字機字體之設計為 typewriter。

文稿中任何地方都可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 TEX 指令類似，我們稱之為「標準字體指令」。譬如，英文 text 要改變為粗體字 **text**，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 \TeX 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有對應之簡化指令。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用大括號界定範圍，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：

```
Start with Roman, {\sl switch to slant}, back to Roman.
```

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。

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

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

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。以上之指令改變字體序列與字形，選用字體族之方法，請見第 16 章之說明。

\TeX 的宣告指令如果下於大括號範圍內，指令之效力僅限於大括號內。 \TeX 提供許多指令環境，譬如，排版表格可使用 `tabular` 指令環境。宣告字體指令如果置於 `tabular` 指令環境內，指令環境結束之後，字體指令之效力也消失。舉例來說，如果表格要以斜體字排版，則在 `tabular` 指令環境開始之處下 `\it` 宣告指令；表格排版結束之後，即自動回復正常字體。

<code>\ctxfml</code>	石創際手祝迴	<code>\ctxfrl</code>	石創際手祝迴
<code>\ctxfm</code>	石創際手祝迴	<code>\ctxfr</code>	石創際手祝迴
<code>\ctxfmb</code>	石創際手祝迴	<code>\ctxfrb</code>	石創際手祝迴
<code>\ctxfmu</code>	石創際手祝迴	<code>\ctxfru</code>	石創際手祝迴
<code>\ctxfmx</code>	石創際手祝迴	<code>\ctxfrx</code>	石創際手祝迴
<code>\ctxfbl</code>	石創際手祝迴	<code>\ctxffl</code>	石創際手祝迴
<code>\ctxfb</code>	石創際手祝迴	<code>\ctxff</code>	石創際手祝迴
<code>\ctxfbb</code>	石創際手祝迴	<code>\ctxfkl</code>	石創際手祝迴
<code>\ctxfbu</code>	石創際手祝迴	<code>\ctxfk</code>	石創際手祝迴
<code>\ctfbbx</code>	石創際手祝迴	<code>\ctxfku</code>	石創際手祝迴
<code>\ctxf1</code>	石創際手祝迴	<code>\ctxfkx</code>	石創際手祝迴
<code>\ctxf1b</code>	石創際手祝迴		

圖 6.1: 中文字體範例

在傳統的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：

正常排版與加底線。

正常排版與`\underline{加底線}`。

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

6.2.2 中文字體

上一小節所介紹之字體指令只能變換英文及數字之字體，無法變更中文字體。中文字體的變換必須利用 `cwTeX` 之字體指令。在 `cwTeX` 第2版以前，中文字體指令同時選用字體與大小，例如，`\m12` 選用中文明體 12 點 (12pt) 字體。2.1 版開始，透過 \LaTeX 之新式字體選用法 (NFSS)，排版者可以僅選定字體種類，字體大小則由 \LaTeX 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，文稿內之中英文字不會有大小不一的情況出現。

圖 6.1 列出 `cwTeX` 現有的中文字體及指令。上一小節說明，英文字體可區分為字形，字體序列，與字體族，但中文字體並無類似的分類。因此，選用字體時，我們直接鍵入字體名稱即可。譬如，中明體指令為 `\ctxfm`，粗明體指令為 `\ctxfmb`。同理，中文黑體指令為 `\ctxfb`；粗黑體指令為 `\ctxfbb`。文稿內若未下任何中文字體指令，`cwTeX` 將自動選用明體字。欲在行文中變更字體，指令如下：

... 總督府推動{\ctxfbb 資本主義化}的政策, ...

「資本主義化」5 個字將改為粗黑體。



舊式指令

若新舊兩種字體指令夾雜使用, 文稿可能會出現中文字大小不一的情況。如果你的文稿檔案內仍使用舊式字體指令, 我們強烈建議更新為新式指令, 方法如下。以文字編輯軟體將文稿內所有的 `\m10`, `\m11`, `\m12` 等指令替代為 `\ctxfm`。同理, 將所有的 `\bb10`, `\bb11`, `\bb12` 等指令改為 `\ctxfbb`。

`ctWTeX` 中文字體指令之有效範圍與 `TeX` 英文字體宣告指令相同, 若字體指令下於大括號或指令環境內, 指令僅於該範圍內有效。上例中, 若原先是使用明體字, 大括號之內之 5 個字將改以粗黑體排版; 但之後又回到明體字。以下的例子說明指令環境對於字體變更之效果。

鄭愁予的詩

我打江南走過,
那等在季節裡的容顏
東風不來, ...

傳誦江湖, 歷久不衰。...

鄭愁予的詩

```
\begin{verse}\ctxff  
我打江南走過, \\  
那等在季節裡的容顏 \\  
東風不來, \dots{}  
\end{verse}  
傳誦江湖, 歷久不衰。 \dots{}
```

在 `verse` 指令環境之前, 中文是以明體字編排。 `\begin{verse}` 指令之後加入 `\ctxff`, 故詩文是以仿宋體排版, 但之後又回到明體字。

行文中改變字體必須以左右大括號界定其範圍。有時候, 我們須在 `TeX` 指令範圍內變換中文字體。譬如,

```
\section{{\ctxfk 耕者有其田}}
```

外圈的大括號是 `\section` 指令的一部分, 內圈的大括號是中文字體更動的範圍。在此情況下, 指令可簡化為:

```
\section{\ctxfk 耕者有其田}
```

圖 6.1 共計列出 23 種字體, 但每一種字體尚可作橫向縮小或傾斜之變形。例如, `\ctxfbb` 可選用粗黑體, `\ctxfbbs` 指令選用粗黑斜體字, `\ctxfbbe` 為粗黑狹長字體, 而 `\ctxfbbes` 則選用粗黑狹長斜體字。變形字體之例子請見表 6.3。同理, 選用狹長明體字之指令為 `\ctxfme`; 楷斜體之指令為 `\ctxfks`。中文變形字

表 6.3: 中文字體之變形

標準	標準	標準	標準
<code>\ctxfbb</code>	<code>\ctxfbbs</code>	<code>\ctxfbbe</code>	<code>\ctxfbbes</code>

之式樣, 如傾斜角度與水平縮小比例, 都可以自行設定, 請見 16.3 節 (頁 272) 之說明。

以上的字體是用於橫排之文稿。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 `examp7.ctx`, 置於 `c:\texmf\cwte\examples` 子目錄內。不過, \TeX 系統是用於排版橫排文稿。若用於直排, 則文稿頁面設計之細節, 如頁眉與頁足等, 尚有待進一步測試。

6.2.3 設定全文之中文字體

中文稿件之正文通常選用最易閱讀之明體字排版, 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 \TeX 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 提供設定全文特定段落中文字體之指令。

如第 2 章例 2 (頁 24) 所示, 欲設定文稿 `\section` 與 `\subsection` 標題內使用圓體字, 可在全文設定區下指令:

```
\ctxfdef{\section}{\ctxfr}
\ctxfdef{\subsection}{\ctxfr}
```

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

```
\ctxfdef{\chapter}{\ctxfbb}
```

中文字體指令 `\ctxfdef` 尚可加入選項以控制排版目錄與頁眉之字體, 請見 7.5.4 節 (頁 100) 之說明。

除了設定 \TeX 指令之字體外, 我們也可以全文設定指令環境 (environment) 之中文字體。上一小節的例子以 `verse` 指令環境排版新詩時, 直接在指令環境內加入 `\ctxff` 指令以選用仿宋體。更好的作法是作全文設定, 如下例所示。

鄭愁予的詩

我打江南走過，
那等在季節裡的容顏
東風不來，...

傳誦江湖，歷久不衰。...

```
\ctxfdef{verse}{\ctxff}  
...  
鄭愁予的詩  
\begin{verse}  
我打江南走過，\\  
那等在季節裡的容顏 \\  
東風不來， \dots{}  
\end{verse}  
傳誦江湖，歷久不衰。 \dots{}
```

若一篇文稿內有幾個地方使用 `verse` 指令環境，則全文設定中文字體可避免前後不一致的情況出現。

上面所介紹的 `verse` 是 \TeX 所定義的指令環境，我們可以自行定義指令環境或巨集指令。而巨集指令或指令環境內尚可以包含其他的巨集指令。譬如，我們可以定義一巨集指令，取名為 `\mymacro`，其中引用另一個巨集指令。`\ctxfdef` 指令之功能類似 \TeX 之巨集指令，但這是由 c\TeX 所提供之指令，使用上有一些限制。特別須注意的是，我們自行定義定集指令或指令環境時，其中不能引用 `\ctxfdef` 指令。以上面的例子而言，`\ctxfdef` 指令不能置於 `\mymacro` 之內。有關於 \TeX 巨集指令之定義方法，請見 14 章 (頁 245) 之說明。

6.3 改變字級

除了選用不同的字體之外，我們也可以變更字體之大小。譬如，章節標題須使用較大的字體排版。以英文字體而言，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 \TeX 原始之字體設計，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，為什麼字體數為 10 點，10.95 點，12 點，14.4 點，...，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，在 \TeX 系統內，當我們說使用 14 點 (或放大 2 級) 之字體時，實際上我們指的是 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。表 6.4 列出字體級數與點數對照表。

表 6.4: 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 $1.2^{0.5} \cong 1.095$ 。

6.3.1 相對大小之字級指令

選擇英文字級有兩種方法, 一種是直接選定字體點數; 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。所謂「相對大小」, 是指比現正在使用之字體放大或縮小。文稿正文使用之標準字體大小, 稱為 `\normalsize`。從標準大小放大 1 級, 指令 `\large`, 放大 2 級 `\Large`。反之, 要縮小 1 級, 指令為 `\small`。表 6.5 列出所有的相對大小指令。

那麼, 文稿之標準字體是如何決定的? 這是由 `\documentclass` 指令所設定。例如以下指令選用 12 點字體為排版正文之標準字體

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

以上指令中, 若將 12pt 選項改為 11pt, 則標準字體大小即為 11pt。反之, 若不入字體選項, 標準字體大小即為內定之 10pt。

相對大小指令所選用之字體, 其點數並不是固定的, 而是依正文標準字體而定。如果我們選定標準字體為 10 點, 則 `\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。如果文稿原選用了較大之字體, 則 `\normalsize` 指令可以變回內定的標準字體。

表 6.5: 相對大小之字級指令

\TeX	\TeX	\TeX	\TeX	\TeX
<code>\tiny</code>	<code>\scriptsize</code>	<code>\footnotesize</code>	<code>\small</code>	<code>\normalsize</code>
\TeX	\TeX	\TeX	\TeX	\TeX
<code>\large</code>	<code>\Large</code>	<code>\LARGE</code>	<code>\huge</code>	<code>\Huge</code>

6.3.2 行距

選用較大點數之字體時，須同時選用較大的行距，版面才會美觀。L^AT_EX 每一種英文字體大小指令有對應之行距。例如，選用 12pt 選項排版正文時，行距有內定值。行文中若使用 `\large` 指令放大字級，行距也隨之而加大。因此，排版英文稿時，我們幾乎不須擔心行距之問題。

但是，中文字之筆劃較英文字複雜，故中文稿之行距須加大一些。若以內定之英文行距排版中文，版面上之中文字看起來會擠成一團。欲更改全文之行距，有幾個方法。譬如，若行距要放大為內定值之 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}
```

以上之指令把全文之行距通通放大為原來的 1.15 倍。文稿中若有註解，其行距也會同比率加大。但是，註解是以較小之字體編排，故行距之放大倍數應較正文小一些。若要進一步微調註解之行距，請見 8.6 節（頁 123）之說明。

以上指令把行距放大為某一倍數。如果全文之行距要改為某一數值，可使用下列指令：

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

以上指令將全文行距一律變更為 15pt。

6.3.3 同時變更字級與行距

另外一種改變字體點數的方法是使用 NFSS 之 `\fontsize` 指令。此一指令同時變更字體點數與行距。假設要選用 14.4 點字體，而行距要變成 20 點，指令為：

```
\fontsize{14.4}{20pt plus .4pt minus .3pt}\selectfont
```

第一個大括號內之 14.4 選用 14.4 點字體；第二個大括號內之數字選定行距為 20 點，其後為行距調整之伸縮彈性正 0.4 點與負 0.3 點。

表 6.6: 選用字體與行距之巨集套件: mymacro.tex

```

\newcount\fs
\def\siz#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\fi}

```

排版時，我們須選定版面文字之高度。因此選擇行距時，我們同時也決定每一頁的版面上要排入多少行文字。譬如，如果選擇版面高度為 310 點，而行距為 20 點，則版面上可排入 15 行，第 16 行擠不進去。但是，行距指令中若加入伸縮彈性，則實際之行距可以因應版面高度而略作改變，這有助於調整出最佳行距。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。

排版中文稿時，使用 `\fontsize` 指令的目的通常不在改變字體點數，只是要變更行距。在中文稿中，若正文選用 12pt 字體，我們可在 `\begin{document}` 指令之後加上 `\fontsize{12}{20pt}\selectfont` 一行指令，將行距改為 20 點。排版一般的短文，幾乎沒有必要再改變字體與行距。但是，如果是排版較複雜的書稿，則可能有必要改變行距。譬如，文稿內之表格若使用小字體排版，行距也須對應調整，此時，使用巨集指令較為方便。

表 6.6 以 `\fontsize` 定義幾組各種字體點數與行距之巨集套件。各巨集指令組合成巨集套件 `mymacro.tex`，存放於 `c:\texmf\cwtex\examples` 檔案夾內。此巨集套件是以 \TeX 指令簡單組合而成，使用方法與 \LaTeX 之巨集套件稍有不同。欲使用此巨集套件，請在全文設定區加入下列一行指令：

```
\input mymacro
```

文稿內要選用 17.28 點之字體（行距設定為 22 點），只要鍵入 `\siz17` 即可。若要選用 11 點之字體（行距為 17 點），則鍵入 `\siz11`。

6.3.4 選用任意點數之字體

Knuth 以 `Metafont` 程式所設計之 `Computer Modern` 英數字體僅有特定之點數。不過，`Computer Modern` 字體已經被轉換成 `Type 1` 描邊字型格式，因此，我們

可選用任意點數之字體。欲選用任意點數,若 13.5 點,我們須在全文設定區引用 type1cm 巨集套件 (作者為 David Carlisle):

```
\usepackage{type1cm}
```

引入此巨集套件之後,即可透過 \fontsize 指令選擇任意之字體點數與行距。

6.4 選擇字級與行距

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。

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，這個比率只供參考，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，行距之比率須加大一些。另外一個重要的考慮因素是行長。一行越長，行距也須加大。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與英文不同。cwTeX 的英文與數字直接取用 T_EX 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。

L^AT_EX 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。綜合以上所述，排版時應避免的錯誤列舉如下。

- 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 L^AT_EX 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（文字版面寬度）也有關係。版面太寬時，閱讀壓力也上升，故行距須加大。中文稿行距沒有標準數值。如果是一般文稿，正文選用 12 點字體，行距請試用 18–20 點。本書正文選用 10.95 點字體，行距則設為 17.8 點。



行距

- 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，行長變成 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 `cmTEX` 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

章節標題之字體必須顯目；正文之字體則是要易讀。我們偶而可看到一些以楷體、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。

7 文稿結構

使用 \TeX 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短文各部分的組成方式。一本書可能包含題目頁、版權頁、目錄、序言、章與節、附錄、索引等。相對而言, 一般的短文通常只有題目、作者、日期、節與小節、參考文獻等。就題目而言, 書籍的排版方式與短文之排版方式不同。書籍通常把題目單獨排為一頁, 短文則常把題目排版於第一頁的上方。

動手排版之前, 我們首先須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, \TeX 即以內定之格式排版。如果 \TeX 所提供之指令不能滿足你的需求, 我們還可以使用各種巨集套件 (package) 進一步調整文稿結構之設計。

前兩章已簡單說明排版指令的意義。巨集指令是把多個排版指令集合在一起, 以進行某項排版工作。譬如, \TeX 排版節之指令為 `\section`。此一看似簡單之指令須完成下列動作: 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, 準備排版下一段文字。以上每一個動作可能都需要數個指令, 為方便起見, \TeX 把以上之指令集合在一起, 並命名為 `\section`, 這就是一個巨集指令 (macro)。進一步而言, 我們可以把性質相關的巨集指令組合在一起, 這就是所謂的巨集套件。

每本書的內容不同, 因此各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, 只要修改巨集指令之定義, 即可一舉改變所有例子之字體。

如果是排版一般的論文, 則 \TeX 所提供之指令已足敷所需。但如果你對於排作品質要求較高, 則花一點時間了解某些巨集套件之功能, 將會大幅提升排作品質。本書除了說明 \TeX 的指令之外, 也將介紹一些常用的巨集套件。

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

7.1 長度單位

設定版面的寬度與高度、訂定行距等，須使用長度的單位。字體大小也是以長度單位衡量。字體設計家在設計字體時，都設想實際排版時最美觀的尺寸，稱為設計尺寸 (design size)。 \TeX 的英文及數學符號字體，大部分的設計尺寸都是 10 點 (printing points, 簡稱爲 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。爲了與英文字配合， $\text{c}\omega\text{T}\text{E}\text{X}$ 中文字體的設計尺寸也都是 10 點。專業排版常以 printing point 作爲長度的單位，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=12cm
```

即將文字版面之寬度訂爲 12 公分。事實上，設定長度的標準指令是 `\setlength`。若使用標準指令，上一行應改爲：

```
\setlength{\textwidth}{12cm}
```

以上所設定的長度稱爲固定長度 (fixed length)，有時候我們必須設定彈性長度 (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底下還有一點空白，但卻不夠再塞進一行。如果行距設爲彈性長度， \TeX 可以把行距縮小一些，以便多排入一行，底下是一個例子：

```
\setlength{\textwidth}{12cm}
\setlength{\baselineskip}{18pt plus0.3pt minus0.2pt}
```

第 2 行指令設定行距爲一彈性長度，`\baselineskip` 爲行距之指令，18pt 爲行距之正常大小，`plus0.3pt minus0.2pt` 爲彈性大小。因此，行距最大是 18.3pt，最小是 17.8pt。`\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，標準行距將縮減 5pt。

有時候我們須在文稿內產生水平或垂直空白。垂直空白的指令為 `\vspace`；相對而言，橫向空白之指令為 `\hspace`。譬如，要在版面中間空出垂直 3 公分的空白，指令為 `\vspace{3cm}`；同理，要在某行文字內插入 2 公分的空白，指令為 `\hspace{2cm}`。請注意，`\vspace` 指令若恰好是下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。

不管是 `\vspace` 或 `\hspace`，都是在大括號內填入某一長度數值，這是所謂的強制變數 (mandatory argument)。有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，選項變數則加於方括號內。顧名思義，選項變數可加可不加。譬如，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：。反之，若要把直線下移 0.1 公分，指令為 `\rule[-1mm]{1cm}{0.1cm}`。

7.1.1 版面大小

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，一旦選定紙張種類，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[a4paper]{article}
\begin{document}
...
\end{document}
```

`\documentclass` 稱為「文稿類別指令」，這通常是文稿的第一道指令。本例使用 `article` 文稿類別排版短文。若是排版書籍，我們可使用 `book` 文稿類別。

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

第 2 行 `\begin{document}` 指令與最後一行的 `\end{document}` 是成對出現的，兩道指令之間即為文稿內容。在 `\end{document}` 指令之後的任何文字與指令，`latex` 程式都不處理。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。`TeX` 提供許多指令環境，譬如排版表格可使用 `tabular` 指令環境。

版面大小與列印之紙張關係密切。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，`TeX` 自動選用 `letter size`，其大小為 `8.5 × 11` 英吋。

要選用 `A4` 紙張，只須在 `\documentclass` 指令中加入選項即可。紙張選項如表 7.2 所示。因此，若選用 `A4` 紙張，正文字體以 `12pt` 排版，指令為：

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

指令之後的方括號內為選項，若不加選項，`TeX` 將使用內設值。因此，若去掉所有的選項，指令變成：

```
\documentclass{article}
```

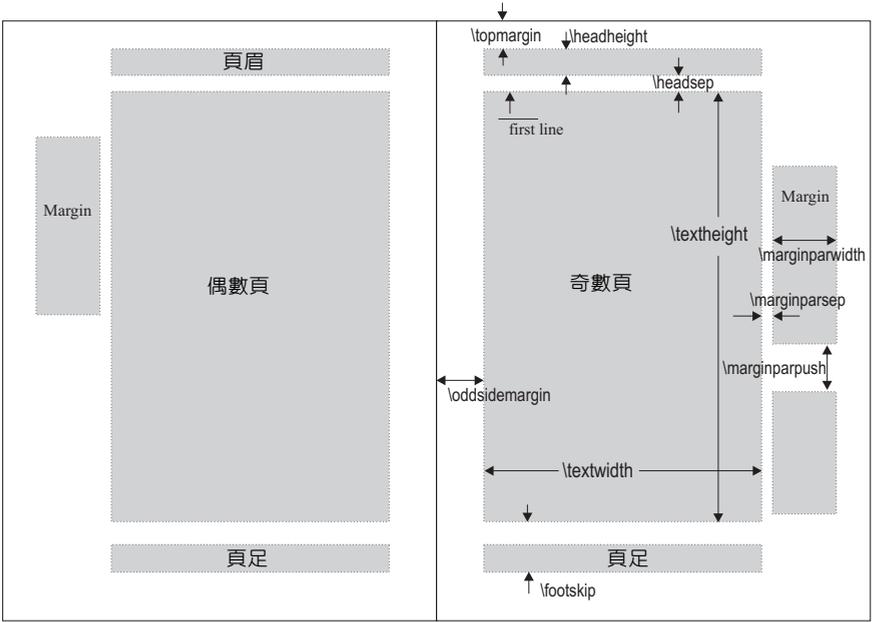
`TeX` 將選用 `letter size` 紙張，並以 `10pt` 字體排版。

短文常列印在 `A4` 紙張上，書籍則有不同之尺寸。因此，排版書籍時須自行設定紙張大小。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)。以本書為例，紙本大小是 `17.0 × 25.5` 公分，故我們在全文設定區以下列指令設定：

```
\paperheight=25.5cm\paperwidth=17.0cm
```

以上兩道指令若分為兩行輸入變成：

```
\paperheight=25.5cm
\paperwidth=17.0cm
```



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

圖 7.1: \LaTeX 之版面與控制指令

效果相同。紙張大小選定之後，版面正文方塊之大小即隨之而調整。大多數的排版結果是列印於垂直紙面上，這稱為 `portrait` (畫像) 模式，也是 \LaTeX 內定之模式。但若在文稿類別指令中加入 `landscape` 選項，則文稿將以橫向或所謂的 `landscape` (風景) 模式排版。

文稿的每一頁除了正文文字之外，還有頁足、頁眉、邊註等。如圖 7.1 所示，正文方塊是指 `\textheight` 與 `\textwidth` 所形成之長方形。排版於正文方塊上方

之文字稱為 header, 傳統中文排版稱之為「天頭」, 本書將稱之為「頁眉」。相對而言, 正文方塊下方之資訊稱為 footer, 傳統排版稱之為「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, 本書將稱之「頁面格式」。正文方塊的大小、正文與頁眉的距離、邊註的寬度等, 都有內定值, 但也都可以重新設定。如果要改變正文方塊大小, 可在全文設定區加入下列指令:

```
\textwidth=11.5cm
\textheight=20.5cm
```

以上指令重新設定正文方塊為 11.5 公分寬, 20.5 公分高。

我們雖然重新設紙張大小, 但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點為列印的基準點。基準點通常距離紙張上沿及左沿各一公分。若正文方塊的寬度及高度是自行設定, 列印之後正文方塊四周的空白可能並不對稱。此時, 我們可以利用 `\voffset` 與 `\hoffset` 指令調整基準點。例如, 要將基準點上移 0.5 公分, 右移 1 公分, 我們可在全文設定區加入下列設定指令:

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

7.2 文稿結構

上一節說明, 我們可使用 `article` 文稿類別排版短文, 複雜的長篇文稿或書籍可使用 `book` 文稿類別。此外, 我們還可使用 `report` 報告文稿類別, 其章節結構比 `article` 複雜, 但比 `book` 簡單一些。以上之文稿類別可以靈活應用。若書籍之內容單純, 我們可以直接使用較簡單的 `article` 文稿類別排版。反之, 若文章的內容複雜, 則以 `book` 文稿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` 文稿類別, 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。

圖 7.1 列出單雙頁並排之版面。排版書籍時, 雙數頁為左頁, 單數頁為右頁; 而且左右兩頁之版面會稍向中間靠攏。這是 `book` 文稿類別之標準版面。相對而言, 若以 `article` 文稿類別排版短文, 每一頁之版面都是在同一位置。

7.2.1 指令選項

大部分的 \LaTeX 之指令都可以加入選項 (options)。舉例來說, `\documentclass` 文稿類別指令除了使用 `article` 文稿類別之外, 也可以加上 12pt 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。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。

上面說明,若選用 book 文稿類別,單雙頁之版面將向書脊靠攏。換言之,紙面上單數頁的版面會較靠近左邊;雙數頁的版面則較靠右邊,如本書之格式。依 \LaTeX 之設定, article 與 report 文稿的單雙頁版面都是位於紙面同樣位置。若希望單雙頁版面也是向書脊靠攏,可在文稿類別指令中加入 twoside 選項。以 book 文稿類別排版書籍時,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,文稿類別指令中應加入 openany 選項。

本書第 9 章將介紹數學式之排版。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,應在文稿類別指令中加入 leqno 選項。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 $\backslash\text{mathindent}$ 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,a4paper]{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,文稿類別指令常用之選項如下:

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 8.7 節之說明。

7.2.2 巨集套件

表 7.3 之文稿類別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。各國的 \LaTeX 專家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, \LaTeX 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 $\backslash\text{usepackage}$ 指令引用之,

```
\documentclass[11pt,a4paper]{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,a4paper]{article}
\usepackage{fancyhdr}
\usepackage{array}
```

或者

```
\documentclass[11pt,a4paper]{article}
\usepackage{array,fancyhdr}
```

請注意, 各選項之間不得留有空白。

以上所介紹的文稿類別指令名為 `\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稿類別指令之選項。如果你仍使用舊版 $\text{\TeX}2.09$, 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。

7.3 文稿類別

大部分的文稿都是以 `article` 或 `book` 文稿類別排版而成。譬如, 學術論文以 `article` 排版, 教科書則以 `book` 排版。碩士與博士論文之排版用任何一種文稿類別都可以。本節將以一實際論文為例, 分別使用兩種文稿類別排版, 以作比較。不過, 我們首先說明 `article` 文稿類別之功能與格式。

第 2 章提供幾個以 `article` 排版短文的例子。在例 2.2 (頁 24) 中, 第 1 行即為文稿類別指令 `\documentclass`, 使用 `article` 編排文稿, 並選用 12pt 字體。第 9 行 `\begin{document}` 指令開始即進入 `document` 指令環境, 其下為文稿內容, 文稿最後一行為 `\end{document}`。本例子之檔案名為 `examp2.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題目、作者姓名、日期等，這是由6–8行指令所排版。第6行以 `\title` 指令排版題目，第7行以 `\author` 指令排版作者名字。第8行之 `\date` 指令若省略， \TeX 將自動排出當天的日期。第10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` 文稿類別下，題目之下即接著排版正文內容；但在 `book` 文稿類別下，題目將獨占一頁。第11行以 `\fontsize` 指令將行距拉大為20點，但字體仍為12點。

正文之節與小節是以 `\section` 與 `\subsection` 指令編排， \TeX 會自動編上號碼。本例計分3節，都是以 `\section` 指令編排。節指令除了自動編號之外，還會選用較大之字體。本例以 `\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44行之 `\subsection` 指令編排小節標題，因為位於第2節之內，因此自動編號為 2.1。小節之中文標題也是選用圓體字。

7.3.1 `article` 文稿類別

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，或更大的字體，編排正文。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。結果當然是慘不忍睹！最常見的問題是：每一行太長，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 \TeX 排版研究生論文，我們可使用 `article` 文稿類別，或者 `book` 文稿類別。本小節將介紹一個以 `article` 文稿類別排版碩士論文的例子，下一小節則將同一論文改以 `book` 文稿類別編排，以資比較。在 `c:\texmf\cwtex\examples\thesis` 子目錄內有 `thesis1.ctx`，這是由一篇實際的碩士論文稍加編輯而成。全篇論文的結構見圖 7.2，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` 文稿類別排版。
- 2–4行：引入巨集套件，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，並加入 `small` 選項以略為縮小標題文字之字級。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行指令之作用。
- 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”。

```

1 \documentclass[12pt,a4paper]{article}
2 \usepackage[sf,small]{titlesec}
3 \usepackage{array,dcolumn,psfrag,graphicx}
4 \usepackage{booktabs,lscape}
5
6 \hoffset=1cm\voffset=-1cm
7 \textwidth=12.5cm \textheight=22cm
8
9 \ctxfdef{\section}{\ctxfbb}
10 \ctxfdef{\subsection}{\ctxfbb}
11
12 \renewcommand{\figurename}{圖}
13 \renewcommand{\tablename}{表}
14 \renewcommand{\contentsname}{\ctxfbb 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 \ctxfk
46 感謝嘉南農田水利會 ...
47 \end{center}
48
49 \newpage
50 \thispagestyle{empty}
51 \begin{center}\Large \ctxfbb
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 \ctxfbb
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 嘉南大圳以遼闊之灌溉面積 ...
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. (1933), ...
96
97 \leref 三浦博亮 (1923),
98 〈嘉南大圳と輪作方式〉, ...
99
100 \end{document}

```

圖 7.2: article 文稿類別

- 16–17行: 定義一指令 `\lref`, 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: 排版獻詞 (dedication)。
- 49–57行: 排版感言與謝詞。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。L^AT_EX 會自動將 `\section` 及 `\subsection` 等指令編排之標題排入目錄內, 但目錄上方會加上 **Contents** 英文字。第 14 行指令的目的則是將 **Contents** 字改為中文「目錄」兩字, 並選用粗黑體。但是, 標題若不是以 `\section` 或 `\subsection` 指令編排, 則標題文字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, L^AT_EX 不自動編號, 也不納入目錄內。我們必須以 `\addcontentsline` (第 89, 93 行) 指令將標題編入目錄。指令中之 `section` 選項表示目錄內之格式與節標題相同。
- 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 (第 77 行) 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; 方括號內為指令選項, 可以省略不加。若不加入選項, 則目錄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, `latex` 即以中括號之內容排版目錄。有時候, 章節之標題相當長, 不適合全部納入目錄內。此時, 方括號內可輸入較簡短之標題, 目錄即以簡短標題排版。

以 `book` 文稿類別排版時, 頁眉上會自動排版章節標題。`\section` 指令中若加入方括號選項, 則頁眉之排版也是取自方括號內之文字。本例之第 9 行以 `\ctxfdef` 指令設定節標題使用中文粗黑體編排, 但並未設定方括號選項之字體, 故目錄以內定之明體字排版。如果目錄要使用其他字體, 可在第 9–10



目錄

表 7.3: 文稿類別

	article	book	report
題目	第 1 頁上方	自成一頁	自成一頁
章節	<code>\section,</code> <code>\subsection,...</code>	<code>\chapter, \section,</code> <code>\subsection,...</code>	<code>\chapter, \section,</code> <code>\subsection,...</code>
版面位置	單雙頁相同	單雙頁往中間靠攏	同 article

行指令內加入選項。例如, 若目錄與頁眉之中文要改用仿宋體, 第 9 行指令應更改為:

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

- 93–99 行排版參考文獻, 採用 hanging indent 格式排版。L^AT_EX 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 L^AT_EX 之設計, 在 article 文稿類別下, 節標題並不新起一頁。本篇論文計分 4 節, 都是以 `\section` 指令排版。我們在指令之前加上 `\newpage` (72 行) 以新起一頁。

7.3.2 book 文稿類別

上一小節的例子以 article 文稿類別排版論文, 同樣的論文也可以使用 book 文稿類別排版, 但部分指令須稍加修改。以 article 文稿類別排版時, 整篇論文分為 4 節, 若改以 book 文稿類別排版, 則分為 4 章。換言之, 原先之 `\section` 指令應改為 `\chapter`; `\subsection` 指令則改為 `\section`。

表 7.3 為文稿類別的簡單比較。除了章節之結構不同之外, 題目頁之格式也不一樣。舉例言之, 若選用 book 文稿類別排版, 則以 `\title` 指令排版書名時, 書名與作者將自成一頁。反之, 若是選用 article 文稿類別排版, `\title` 指令會直接將題目排版於第 1 頁上方。

book 文稿類別之頁面格式也與 article 文稿類別不同。book 文稿類別會在每一頁之頁眉 (header) 或頁足 (footer) 自動排出章節標題。我們把 book 文稿類別排版之例子檔案命名為 `thesis2.ctx`, 置於 `c:\texmf\cwtex\examples\thesis` 檔案夾內以供參考。為方便說明, 主要的修正指令置於圖 7.3。

- 第 1 行: 選用 book 文稿類別。我們另外加上 `oneside` 與 `openany` 兩個選項。依 L^AT_EX 內定值, 以 book 文稿類別排版時, 左頁 (偶數頁) 將往右靠向書脊,

```

1 \documentclass[12pt,oneside,openany,a4paper]{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[R0,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 7.3: book 文稿類別

右頁則往左靠向書脊。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` 文稿類別排版時，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 17.5.2 節 (頁 302)。L^AT_EX 本身已提供排版頁眉之指令，但無法處理中文。故我們引入 `fancyhdr` 巨集套件排版頁眉。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，故我們須以 `fancyhdr` 之指令重新設計頁眉之排版，請參見 `thesis2.ctx` 檔案內之指令。
- 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。指令之說明請參考 17.3 節 (頁 286)。
- 19–20 行: `cwLATEX` 字體指令，選擇以粗黑體排版章與節之標題。

就本例子而言,使用 `article` 文稿類別與 `book` 文稿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「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` 文稿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,不同的文稿類別可以排出相同的版面。當然,若文稿甚長時,使用 `book` 文稿類別仍有其便利之處。第 19 章將有進一步的說明。

7.4 題目與摘要

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` 文稿類別排版時,題目將獨占一頁;但在 `article` 文稿類別中,題目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`,如第 2 章例 2.2 (頁 24) 所示,各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 `cwTeX` 指令設定。

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。第 2 章例 2.2 中,若嫌日期之字體太大,我們可以用 `\small` 指令縮小之,例如:

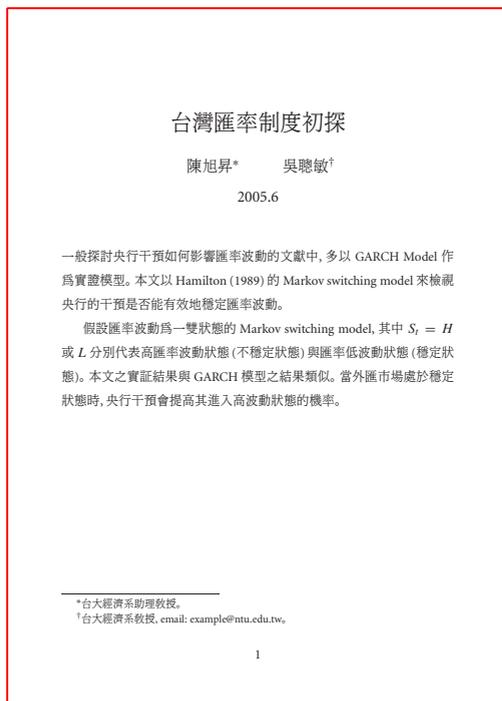
```
\date{\small 2005.1.4}
```

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,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。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{陳旭昇·吳聰敏}
```

事實上,這與單一作者的排版並無不同,我們只是在兩位作者名字中間加上一小點。相反的,例 7.1 之排版就稍為複雜一些,排版指令是在第 7–11 行。兩位作者



```

1 \documentclass[12pt,a4paper]{article}
2 \usepackage{amsmath}
3 \usepackage[sort]{natbib}
4 \setlength{\bibsep}{2pt}
5 \renewcommand{\refname}{\ctxfbb 參考文獻}
6 \title{台灣匯率制度初探}
7 \author{陳旭昇\thanks{
8 台大經濟系助理教授。}\and
9 吳聰敏\thanks{台大經濟系教授,
10 email: example@ntu.edu.tw.}}
11 \date{2005.6}
12 \begin{document}
13 \maketitle
14 \fontsize{12}{20pt}\selectfont
15 \noindent
16 一般探討央行干預如何影響匯率波動的文獻中,
17 多以 GARCH Model 作為實證模型。
18 本文以 Hamilton (1989) 的 Markov switching model
19 來檢視央行的干預是否能有效地穩定匯率波動。
20
21 假設匯率波動為一雙狀態的 Markov switching model,
22 其中  $S_t=H$  或  $S_t=L$  分別代表高匯率波動狀態
23 (不穩定狀態) 與匯率低波動狀態 (穩定狀態)。
24 本文之實証結果與 GARCH 模型之結果類似。
25 當外匯市場處於穩定狀態時,
26 央行干預會提高其進入高波動狀態的機率。
27
28 \end{document}

```

例 7.1: 多位作者與謝詞

名字之後都以 `\thanks` 指令排版其單位; 而兩位作者名字之間則以 `\and` 區隔。請注意, 第 8 行指令末端有一換行指令。

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,

```

\title{Cyrillic and Oriental Typography in Rome at\\
the End of the Sixteenth Century}

```

排版之後, 兩段文字都將居中排列。

一般而言, 書籍之書名與作者名字之排版都經過特別設計, 故以上之指令主要用於 `article` 與 `report` 文稿類別。若自行設計標題頁, 指令與文字內容可置於 `titlepage` 指令環境內:

```

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

```

請注意, 在此指令環境內 `\title` 與 `\author` 等指令皆不能使用, 而指令環境之後也不須下 `\maketitle` 指令。

7.4.1 摘要

文稿正文之前若要排版摘要, 可以使用 `abstract` 指令環境編排。L^AT_EX 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。因此只要在全文設定區重新定義即可:

```
\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` 文稿類別裡, 摘要將排版於標題下面; 在 `report` 文稿類別裡, 摘要自成一頁, 而且不編上頁碼。書籍的摘要通常自成一章, 或放入序文內, 因此 `book` 文稿類別裡不能使用 `\abstract` 指令。

7.5 章節標題

文稿通常區分章節, 因此 L^AT_EX 提供許多編排章節之指令。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 1 章之編號, 中文則使用「第 1 章」或「第一章」。如何編排中文標題, 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, 我們有一個功能甚佳的巨集套件 `titlesec` 可資使用。善用此一巨集套件, 可輕而易舉地重新設計出章節標題。

7.5.1 章節標題之層級

L^AT_EX 排版章節標題之指令如表 7.4 所示。章節標題指令與文稿類別關係密切。若是使用 `book` 或 `report` 文稿類別排版, 從 `\part` 到 `\subparagraph` 都可使用; 但 `\part` 指令為選用, 若不用 `\part`, 即直接從 `\chapter` 開始。相對而言, 若是使用 `article` 文稿類別, 文稿內不能使用 `\chapter` 指令, 但仍可選用 `\part` 指令。若不使用 `\part`, 節層級結構即由 `\section` 開始。章節標題指令有其層級結構, 先後順序不能倒置。

第 2 章的例 2.2 (頁 24) 中, 節標題是以 `\section` 指令編排, 小節標題則是以 `\subsection` 排版。文稿中遇有節與小節標題指令時, 即另起一行, 自動編上號碼, 選用稍大字體, 排出標題文字; 正文內容將排於下一行。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。

表 7.4: 章節指令之層級與編號

章節指令	article	book	report
<code>\part</code> (部)	選用	選用	選用
<code>\chapter</code> (章)	無	編號	編號
<code>\section</code> (節)	編號	編號	編號
<code>\subsection</code> (小節)	編號	編號	編號
<code>\subsubsection</code> (次小節)	編號	不編	不編
<code>\paragraph</code> (段)	不編	不編	不編
<code>\subparagraph</code> (小段)	不編	不編	不編

如表 7.4 所示, 若以 `article` 文稿類別排版, \LaTeX 對 `\section`, `\subsection`, 及 `\subsubsection` 等指令會自動編號。若一開始跳過 `\section` 指令, 直接使用 `\subsection` (小節), 標題仍可排出, 但第 1 小節的自動編號將為 **0.1**, 第 2 小節為 **0.2**, 等等。因此, 除非有特別理由, 章節指令應按順序使用。在 `article` 文稿結構下, 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。

若是使用 `book` 文稿類別, 自動編號之範圍是則從 `\chapter` 到 `\subsection`。因此, 在 `\subsection` 之下, 我們可跳過 `\subsubsection`, 直接使用 `\paragraph` 指令排版小標題。`\paragraph` 指令所排版出來之小標題格式稱為 `running-head`。標題從文字版面之左沿開始排版, 其右會留下一點空白, 接著即排版正文。再下一個層級的 `\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沿。

如果要使用章節標題指令排版, 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 `*` 號標題指令的一個問題是, 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, 請見 7.8 節 (頁 106) 之說明。

書籍或較長之報告若分為數部 (`part`), 每一部之標題可以用 `\part` 指令排版。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, 英文字號與中文標題可能並不搭配。要重新設計部之標題, 可使用 `titlesec` 巨集套件, 詳見 17.3 (頁 286) 節之說明。

以 \LaTeX 的指令排版中文標題時, 有兩個問題須解決。第一是中英文字體之搭配是否適宜, 第二是中英文章節編號之格式不同。以下將說明解決的方法。

表 7.5: 中英文字體之搭配

7.5 章節標題	英文粗體加中文特明體 (\ctxfmu)
7.5 章節標題	英文 sans serif 加中文圓體 (\ctxfr)
7.5 章節標題	英文 sans serif 加中文粗黑體 (\ctxfbb)
7.5 章節標題	英文標準字體加中文仿宋體 (\ctxff)

7.5.2 設定章節標題之字體

\LaTeX 之章節指令會選用英文粗體字, 但這些指令對於中文並無作用, 因此我們須加入指令選擇中文字體。在下指令之前, 我們首先須考慮中英文字體之搭配。表 7.5 列出幾種可能的搭配, 其中第一種是以中文特明體搭配英數字之粗黑體。英文粗黑體筆劃較粗, 若配以中文粗明體, 效果並不理想。

如果中文標題使用圓體或粗黑體, 較理想之搭配應該是 sans serif 字族。但是, \LaTeX 章節指令內定的是粗黑字體 (\textbf)。因此, 如果要選用表 7.5 之字體搭配, 中文與英文字體須同時變更。本小節首先說明變更英數字體的指令, 中文字體之變更請見 7.5.4 節 (頁 100)。事實上, 變更章節標題之英數字體最簡單的方法是使用下一小節所介紹的 titlesec 巨集套件; 此處僅先介紹一般性之概念。

在 \LaTeX 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。



字體搭配

表 7.6: 排版數字之指令

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

利用 `\alph` 指令, 我們可將計數器內之數字改以小寫英文字母排出。因此, 上面第 2–3 行指令若改為:

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

小節編號將變成 1.a, 1.b, ... 等。相對而言, `\Alph` 指令會將阿拉伯數字變成對應的大寫英文字母。如果 2–3 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。以上所介紹的 `\arabic` 與 `\alph` 都是排版數字之指令; 我們將同一性質之指令全部列舉於表 7.6。

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 `titlesec` 巨集套件 (作者 Javier Bezos)。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 (居中或靠左), 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 17.3 節 (頁 286) 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 7.7 列出簡易指令之選項。譬如, 章節標題若原來都是靠左排版, 若要改為居中, 可在全文設定區加入下列兩行指令:

```
\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` 文稿類別, 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。L^AT_EX 之 `\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,

表 7.7: titlesec 巨集套件簡易指令

<code>\usepackage[options]{titlesec}</code>
<ul style="list-style-type: none"> ● 標題字體: <code>rm sf tt md bf up it sl sc</code> ● 標題字級: <code>big medium small tiny</code> ● 標題位置: <code>center raggedright raggedleft</code> ● 上下間距: <code>compact</code>

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*` 指令將標題文字往上移一些。

表 7.7 列出 `titlesec` 巨集套件之簡易指令。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)。若加入兩個以上之選項, 須以逗號分隔。`titlesec` 之字體指令僅對英數字有效, 中文字體須以中文字體指令另行設定。舉例來說, 若中文節標題選用圓體, 而英數字欲改用 `sans serif` 字體搭配, 只要在全文設定區使用下列指令即可:

```
\usepackage[sf,small]{titlesec}
\ctxfdef{\section}{\ctxfr}
```

其中, `small` 選項指示使用放大級數較小之字體。 \TeX 內定之放大字體相當於 `big` 選項。此外, 若加入 `compact` 選項, 則標題與上下文之間距會縮小一些。第 2 行指令選用圓體字排版中文標題, 下一小節將詳加說明。

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。 \TeX 是以 `\thetitle` 指令代表章節之數字編號, 若要在章節編號之前加上 \S 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 \S 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: $\S 7.5$ 。以上指令雖然方便, 但它會更動文稿中全部章節之標題。除非章與節之標題都選用相同的設計, 否則不一定適用。

7.5.4 中文標題之字體

上一小節所介紹的 `titlesec` 巨集套件雖然讓標題之設計調整變得很容易, 但仍然有一個不方便之處, 那就是無法控制標題之中文字體。爲了讓使用者能一舉設定

章節標題之中文字體, cWTeX 提供一簡單的巨集指令: $\backslash\text{ctxfdef}$ 。舉例來說, 如果節與小節之標題要以楷體排版, 只須在全文設定區加入以下指令即可:

```
\ctxfdef{\section}{\ctxfk}
\ctxfdef{\subsection}{\ctxfk}
```

ETEX 提供一排版目錄之指令: $\backslash\text{tableofcontents}$, 此一指令會將文稿內之章節標題排入目錄內。不過, 章節標題可能會選用較顯目之字體, 如粗黑體或圓體, 這些字體並不適合用於目錄內。另一種情況是, 章節標題之文字較長, 而我們希望納入目錄內的是較簡短的標題。為解決以上兩個問題, 在下章節標題指令時, 我們可加入方括號選項, 如下例所示:

```
\section[章節標題]{章節標題}
```

方括號內之文字是用於排版目錄與頁眉之標題。本例之方括號選項與大括號之文字完全相同, 但我們可以透過其他指令選用不同之字體。

若 $\backslash\text{section}$ 指令內加入方括號選項, 則 $\backslash\text{ctxfdef}$ 指令可加入對應之選項, 例如:

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

以上兩行指令中, 方括號內之 $\backslash\text{ctxff}$ 指令設定以仿宋體排版目錄內之節與小節標題。以 book 文稿類別排版時, 章節之標題文字會自動排版於頁眉上。以上 $\backslash\text{ctxfdef}$ 指令中方括號內字體選項, 同時也決定了頁眉上章節標題之字體。

7.6 引述章節

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 (引述)。譬如, 第 5 章內文可能引述 3.2 節的文字或圖 4.1。 ETEX 所提供的 $\backslash\text{label}$, $\backslash\text{ref}$, 與 $\backslash\text{pageref}$ 三道指令即作此用途。 ETEX 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 $\backslash\text{label}$ 指令作標籤, 接下來即可使用 $\backslash\text{ref}$ 或 $\backslash\text{pageref}$ 指令引述。譬如, 本節標題之排版指令為:

```
\section[引述章節]{引述章節}\label{refer}
```

之後在文稿其他地方鍵入下列指令:

... 請參見 $\backslash\ref{refer}\backslash Z$ 節 (頁 $\backslash\pageref{refer}$) 之說明。

排版之後, \LaTeX 會自動引入本節之編號與頁碼:

... 請參見 7.6 節 (頁 101) 之說明。

$\backslash\ref$ 指令引述章節編號, $\backslash\pageref$ 指令則引述頁碼。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, $\backslash\label$ 指令也可以下於數學式之後, 註解 $\backslash\footnote$ 指令之內, 或者圖表標題指令 $\backslash\caption$ 之後。本書有許多圖表, 都是 $\backslash\label$ 指令下標籤, 再以 $\backslash\ref$ 指令引述。譬如, 表 7.8 之標籤指令如下:

```
 $\backslash\caption{\text{設定頁面格式}}\backslash\label{\text{pagestyleop}}$ 
```

引述指令為:「表 $\backslash\ref{\text{pagestyleop}}\backslash Z$ 列出全部的選項。」引述數學式也是利用相同的指令, 請見 9.6.1 (頁 158) 節之說明。

7.7 頁面格式與頁碼

任何文稿幾乎都會排版頁碼, 長篇文稿還會在頁眉或頁足排版章節標題。頁眉與頁足之排版, \LaTeX 稱為頁面格式 (page style)。頁碼通常是排版於頁眉或頁足, 因此是頁面格式的一部分。

\LaTeX 以 $\backslash\text{pagestyle}$ 指令設定頁面格式。譬如, 在全文設定區加入指令:

```
 $\backslash\text{pagestyle}\{\text{plain}\}$ 
```

則頁碼將排版於版面底下之中間, 頁眉則為空白。plain 是 $\backslash\text{pagestyle}$ 指令 4 個可能的選項之一, 表 7.8 列出全部的選項。以 article 文稿類別排版短文時, 通常只加入頁碼, 節與小節之標題並不排於頁眉或頁足上。因此, 我們只須使用 plain 選項即可排出頁碼。不過, \LaTeX 本來就內定使用 plain 之格式, 因此, 我們可以完全省略上述指令。相反的, 如果不想排出頁碼, 則必須在全文設定區加入以下指令:

```
 $\backslash\text{pagestyle}\{\text{empty}\}$ 
```



亂碼

書籍之頁眉或頁足上通常會排版章節之標題。譬如, 本書雙數頁之頁足排版章標題, 單數頁則排版節標題。依表 7.8, 我們可使用 headings 選項以排版頁眉。不過, \LaTeX 內建之指令在排版頁眉之中文標題時會出現亂碼。如果是使用 book 文稿類別, 要避免中文出現亂碼, 請在全文設定區加入以下指令:

表 7.8: 設定頁面格式

<code>\pagestyle{options}</code>	
<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。

```
\pagestyle{headings}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection~ #1}}
```

如果是以 `article` 文稿類別排版, 並使用 `twoside` 選項, 文稿內並無 `\chapter` 指令, 故以上指令之第 2 行應去掉。欲了解以上指令之意義, 請見 17.5 節 (頁 298)。

7.7.1 頁碼

依專業排版之規範, 書籍每一章之標題頁並不排版頁碼。`\pagestyle` 若採 `plain` 選項, 標題頁之頁碼即自動取消。若要取消某頁之頁碼, 須使用 `\thispagestyle` 指令。譬如, 某一頁不想排出頁碼, 可在該頁某處下指令:

```
\thispagestyle{empty}
```

反之, 若每一頁都不排頁碼, 可在全文設定區加入下指令:

```
\pagestyle{empty}
```

以上之指令不僅取消頁碼, 也使頁眉與頁足資訊全部變成空白。

在正常的情況下, 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)。L^AT_EX 利用計數器記錄章節之編號與頁碼數字; 其中, 頁碼是記錄於 `page` 計數器內。欲改變頁碼, 可使用 `\setcounter` 指令:

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

以上指令將頁碼改為 20。

頁碼通常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指令:

表 7.9: 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, ...。

`\pagenumbering{roman}`

其他可用之選項如表 7.9 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`。

有些西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。利用以上之 `\pagenumbering` 指令, 我們可將頁碼改以小寫羅馬字排出。不過, 如果是使用 `book` 文稿類別排版, 我們也可以使用 `\frontmatter` 與 `\mainmatter` 指令達到相同的效果。若在目錄之前加上 `\frontmatter` 指令, 其後之頁碼數字將以羅馬數字編排。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。

除了改變頁碼數字格式之外, `\frontmatter` 指令還有其他效果。以 `book` 文稿類別排版書籍時, `\chapter` 指令可用以編排章標題, \LaTeX 會自動加入編號。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, \LaTeX 仍以一般章標題格式編排, 但不加入數字編號。不過, `\mainmatter` 指令之後, `\chapter` 即恢復自動編號之功能。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。

以上之 `\frontmatter` 指令僅能使用於 `book` 文稿類別。若是以 `article` 文稿類別排版, 我們須自行以 `\pagenumbering` 指令設定頁碼之字體。

7.7.2 換頁

排版時, \LaTeX 會自動選擇適當的地方換頁。如果我們自行在文稿某處加入換頁指令 `\newpage`, 其下的文字將移至下一頁。但是, 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。如果非得空出一頁不可, 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。

我們也可以使用 `\clearpage` 指令換頁。此一指令除了換頁之外, 還會把尚未排出的圖表列印於本頁之後。L^AT_EX 提供浮動圖表指令環境 (float) 以處理圖表之排版。如果文稿內有很多大型的圖表等著排出, 則下了 `\newpage` 指令之後, 這些大型圖表有可能被排版於文稿最末端。為避免這種情況, 我們可利用 `\clearpage` 將這些圖表排於下一頁。

文稿若要排版成兩欄格式, 只要在 `\documentclass` 指令中加入 `twocolumn` 選項即可。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, 下一頁將跳空。

排版書籍時, 是否跳頁常依本頁是單頁或雙頁而定。譬如, 若每一章之標題是排於單數頁, 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:

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

以上的 T_EX 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。

以上的指令解決了是否跳頁的問題。不過, 跳空的那一頁的版面下方仍然會排出頁碼。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與頁足皆為空白, 頁碼自然也不會出現。以上是以 T_EX 指令控制換頁, L^AT_EX 也提供類似的指令, 請見 14.1.3 節 (頁 249) 之說明。

`\clearpage` 或 `\newpage` 指令之作用是強迫換頁。有時候, 本頁剩餘之空白再多加一點就可以容納下一段文字。如果要強迫把下一段文字排入本頁, 可將 `\samepage` 指令加在本段文字末端, L^AT_EX 會試著把下一段文字納入本頁。另外一個辦法是把本頁加長一些。例如, 在本頁當中加入

```
\enlargethispage{5pt}
```

可將本頁之文字版面加長 5pt。

7.8 目錄

TEX 能自動記錄章節與圖表標題之文字內容及其頁碼,因此排版目錄甚為簡單。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 TEX 無法取得所需資訊。

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 2-3 次,章節目錄即出現於指定位置。正文之頁碼通常是以阿拉伯數字編排,目錄之頁碼可能是以羅馬數字編排,也可能是阿拉伯數字編排。若是後一種情況,目錄與正文之頁碼是接續的。因為目錄至少會有幾頁之版面,故必須執行 latex 程式 2-3 次,才能得出正確的頁碼。

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。

根據表 7.4 (頁 97), 章節指令有好幾層。依內定值, book 文稿類別之目錄將排版到 `\subsection` 層級, article 文稿類別之目錄則排版到次小節。欲增加或減少目錄之章節層級,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}
```

詳見 17.3.5 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 17.4 節 (頁 295) 之簡單介紹。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 `cwtext` 程式轉換後的格式,而非原始輸入之中文字。我們可以利用 `cw\TeX` 之工具程式 `tex2xtc` 將之轉回中文字碼。

請先進入 DOS 視窗,再進入 `c:\xtemp` 子目錄內,鍵入底下指令:

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

即產生 `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,即可引入文稿檔案內以排版出自行設計之目錄。

若要排版圖標題之目錄,可使用 `\listoffigures` 指令;同理,排版表格標題目錄之指令為 `\listoftables`。排版之後,工作檔案夾內會產生 `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,請依照上面的作法,使用 `\renewcommand` 指令重新定義 `\listfigurename` 與 `\listtablename` 為中文標題。

7.9 附錄

論文或書籍可能帶有附錄。論文的附錄通常是加在文章末端;書籍的附錄可能是在某章之後。 \TeX 僅對 `book` 與 `report` 文稿類別提供 `\appendix` 指令排版附錄,而且,使用上有一些之限制。簡單來說,`\appendix` 僅能出現於正文所有章節後面。

舉例來說,若在某書末端加入 `\appendix` 指令,之後我們仍以 `\chapter` 指令排版附錄內之章標題。 \TeX 採與正文相同的格式排版附錄之題目,但編號改變。具體之作用如下:

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,章編號將為 B。

台灣匯率制度初探

陳旭昇·吳聰敏*

2005.8

一般探討央行干預如何影響匯率波動的文獻中,多以 GARCH Model 作為實證模型。本文以 Hamilton (1989) 的 Markov switching model 來檢視央行的干預是否能有效地穩定匯率波動。

1 匯率波動

假設匯率波動為一雙狀態的 Markov switching model, 其中 $S_t = H$ 或 L 分別代表高匯率波動狀態 (不穩定狀態) 與匯率低波動狀態 (穩定狀態)。

附錄

A 資料說明

本文實証分析採用 1980:12--2004:12 之月資料, 實質 GDP 則由季資料轉換成月資料。

*email: example@ntu.edu.tw.

```
1 \documentclass[12pt,a4paper]{article}
2 \usepackage{appendix}
3 \renewcommand{\appendixpagenam}{\Large\ctxf 附錄}
4 \title{台灣匯率制度初探}
5 \author{陳旭昇·吳聰敏\thanks{
6   email: example@ntu.edu.tw.}}
7 \date{2005.8}
8 \begin{document}
9 \maketitle
10 \fontsize{12}{20pt}\selectfont
11 \noindent
12 一般探討央行干預如何影響匯率波動的文獻中,
13 多以 GARCH Model 作為實證模型。
14 本文以 Hamilton (1989) 的 Markov switching model
15 來檢視央行的干預是否能有效地穩定匯率波動。
16
17 \section{匯率波動}
18 假設匯率波動為一雙狀態的 Markov switching model,
19 其中  $S_t=H$  或  $L$  分別代表高匯率波動狀態
20 (不穩定狀態) 與匯率低波動狀態 (穩定狀態)。
21
22 \appendix
23 \appendixpage
24 \section{資料說明}
25 本文實証分析採用 1980:12--2004:12 之月資料,
26 實質 GDP 則由季資料轉換成月資料。
27 \end{document}
```

例 7.2: 附錄

若是中文稿件,我們可以在全文設定區下指令,將 **Appendix** 一字改為「附錄」:

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

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, \LaTeX `\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,我們不妨自行設計。例如,若每一章之後都有附錄,我們可使用 `\chapter*` 指令排版,並自行設計版面。有關於版面設計之巨集套件,請見第 17 章之介紹。

\LaTeX 僅對 `book` 與 `report` 文稿類別提供 `\appendix` 指令。若在 `article` 文稿類別內要排版附錄,可引用 `appendix` 巨集套件 (作者 Peter Wilson), 指令與排版結果如例 7.2 所示。第 22 行下 `\appendix` 指令,以下為附錄之內容。第 23 行指令 `\appendixpage` 會排版 **Appendices** 一字。因為是中文稿,我們在第 3 行將英文標題重新定義為「附錄」兩字。附錄之內仍以 `\section` 指令排版節標題,但編號變成 A, B 等。

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、衆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，表示上一段落結束，以下為新段落開始。L^AT_EX 會將此段文字排得整整齐齊。

但是，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，對於排版原始程式碼而言，此一工具非常方便。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。

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。內縮之距離若要拉長為 0.8 公分，請在全文設定區加入下列指令：

```
\parindent=0.8cm
```

如果某一段落之行首不要內縮，必須在段落開始之處加上 `\noindent` 指令。如果是全文之行首都不要內縮，則須把上述之 `\parindent` 設為 0cm。

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。段落之間距與一般行距相同，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，要將段落間距改成 20 點，只要在全文設定區加入以下指令即可：

`\parskip=20pt`

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,不過它可以防止在換行處換頁。另外, `\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。若要左沿對齊,右沿不切齊,可使用 `\raggedright`。反之,若要右沿對齊,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 <code>\\</code>
可靠左	<code>\raggedright</code> 可靠左 <code>\\</code>
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。若右沿對齊, \TeX 須努力調整文字之間距;有時候被迫把文字間距拉得太大,版面反而不美觀。要設定讓右沿不對齊,僅須在文稿前端加入 `\raggedright` 指令即可。

以上三個指令會影響指令以下所有文字段落之編排。如果是要將某幾段文字或圖表居中或靠邊排版,則以指令環境排版較為方便。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, ...	<code>\begin{center}</code> 我打江南走過, <code>\\</code> 那等在季節裡的容顏 <code>\dots{}</code> <code>\\</code> 東風不來, <code>\dots</code> <code>\end{center}</code>
---------------------------------------	---

本例中, `center` 指令環境內全部為普通文字, 但也可以包含其他的指令環境。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 \dots{} \\  
東風不來, \dots  
\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版面垂直方向的中間位置, 我們可以使用 `\vfill` 指令。首先, 在上一段落之後加上 `\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 \TeX 指令, 代表排版文稿的主檔名, 我們須在其後自行加上附加檔名 `.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。上面曾經說明, 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 (約 1 公分)。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，空白可能縮小為 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，指令即失效。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*{1.0cm}` 指令，因此段落間的空白比正常間距多了 1.0 公分。

\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 (約 3.5 公分) 水平空白；`\qquad` 則產生兩倍之距離。

若要直接控制水平間距，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 0.3 公分		距離	<code>\hspace*{-0.3cm}</code>	後退 0.3 公分

若距離值為負數，文字將左移。此例中，第 4 行開頭之 `\hspace{-0.3cm}` 指令讓句子左移 0.3 公分。

仔細比較例子的前三行，可發現句子中間水平空白的大小和 `\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 公分加上左右兩個空白鍵之距離。和 `\hspace` 指令相近的是 `\hspace*`。前一指令若恰好出現在一行開頭處，指令即失效。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，並盡量往左右兩邊靠，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，請見以下的例子。

2005年
2005年

經濟

台大
台大

```
2005年\hfill 台大\  
2005年\hfill 經濟\hfill 台大
```

本例中, `\hfill` 指令事實上等於是 `\hspace{\fill}`。`\hspace` 指令的作用是留出水平空白, 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「經濟」兩個字就從兩邊往中間擠壓。上一節曾介紹居中排版指令 `\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。最後, 如果要畫出一水平線, 應使用 `\hrulefill` 指令。以上的指令中, `\dotfill` 的應用之一是排版目錄, 如下例所示。

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

若使用 `\hrulefill`, 點線將變成實線:

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。如果是要徵引其他文章之段落，可使用 `quote` 或者 `quotation` 指令環境。

8.3.1 `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 \ldots  
\begin{verse}\ctxff  
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 \ldots  
\end{verse}
```

我們在每句之後加上換行指令 `\\`。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。

8.3.2 `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。若引文是中文，我們可以利用 `\ctxfdef` 指令定義特別的字體。例如，

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

即設定 `quote` 指令環境以楷體排版。請見以下例子：

在國慶文告的最後，偉大的領袖一
定會高呼：

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，  
偉大的領袖一定會高呼：  
\begin{quote}  
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。

8.4 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。 \TeX 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。

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， \TeX 會先在各條文之前加上一個圓點，其後再排版條文內容。條列指令環境內還可以使用另一層的條列指令環境，最多可以使用 6 層。若 `itemize` 指令環境下還有一層 `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` 指令環境中，標籤是小寫英文字母 (a), (b), (c) 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， \TeX 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 例子的第 5 行，`\item` 指令若改為 `\item[A]`，排版之後標籤將為：

A 聯盟的組織

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}]`。

工友總聯盟的組織如下：

- 聯盟的組織
 1. 意志機關
 - (a) 聯盟代表大會
 - (b) 各工友會會員大會
 2. 執行機關
 - (a) 執行委員會
 - (b) 各部的組織
- 構成分子
 - 蘭陽總工大會
 - 基隆船炭工大會

```
\leftmargini=7mm
\leftmarginii=6mm
...
工友總聯盟的組織如下：
\begin{itemize}
\item 聯盟的組織
\begin{enumerate}
\item 意志機關
\begin{enumerate}
\item 聯盟代表大會
\item 各工友會會員大會
\end{enumerate}
\item 執行機關
\begin{enumerate}
\item 執行委員會
\item 各部的組織
\end{enumerate}
\end{enumerate}
\end{itemize}
\item 構成分子
\begin{itemize}
\item 蘭陽總工大會
\item 基隆船炭工大會
\end{itemize}
\end{itemize}
```

圖 8.1: 條列指令環境

如果要改變 `itemize` 指令環境之標籤符號，可以重新定義 `\labelitemi` 與 `\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`。譬如，文稿中若使用下列指令：

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

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阿拉伯數字或英文字母依序標出，控制指令有：`\arabic`，`\roman`，`\Roman`，與 `\Alph`。譬如，

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

將把指令環境最外層之標籤改變成 1), 2), ... 之形式。同理，下列指令：

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

則把標籤變成大寫羅馬字: I, II, ...。以上之指令若下於全文設定區,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。

第三種條列指令環境為 `description`。使用此一指令時,必須自行在方括號中輸入標籤,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,並且可以改變字體:

政治 政客玩的遊戲,

政客 玩政治的人,

政治家 英俊的政客。

```
\begin{description}\itemsep=-1pt
\item [{\ctxfbb 政治}] 政客玩的遊戲,
\item [{\ctxfbb 政客}] 玩政治的人,
\item [{\ctxfbb 政治家}] 英俊的政客。
\end{description}
```

其中, `\itemsep=-1pt` 指令的作用是把各列舉項之間距縮小一些。

在 `description` 指令環境下,若某條列項之文字超過一行長度,第2行以下之文字會略往內縮一特定距離。但是在 `itemize` 與 `enumerate` 指令環境下,列舉條文之長度若超過行長時,第2行開始每一行之起頭將內縮於標籤相對位置之後,與第一行之起頭對齊。

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;第二層列舉項則以 `\leftmarginii` 控制。圖 8.1 的例子裡,我們重新設定內縮之距離如下:

```
\leftmargini=7mm
\leftmarginii=6mm
```

8.5 迷你版面與方塊

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， \LaTeX 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。舉例來說，如果我們在某一頁的左上角引用一個外製圖形（見第 11 章），其寬度為正常行長的三分之二，則右邊剩下來的空間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。

下一小節先介紹迷你版面指令，再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` (段落方塊) 指令，或者 `minipage` (迷你版面) 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。因為這兩個指令的功能類似，以下的說明中將簡單以迷你版面稱之，而不作區分。

最簡單的 `\parbox` 指令形式如下：

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

若使用 `minipage` 指令環境，指令如下：

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

以上指令中，`5cm` 設定迷你版面之寬度為 5 公分，`t` 則設定版面上沿對齊基線 (baseline)。所謂基線，簡單來說是指貼著一行文字底部之無形直線。

排版時，迷你版面所形成之方塊不會被拆開。換言之，迷你版面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。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。

網球代表隊員名單如下：

天龍:	柯達	地虎:	張德培
	韓曼		里歐斯

網球代表隊員名單如下：`\\[4mm]`

```
天龍: \parbox{1.2cm}{  
 柯達\\ 韓曼}  
\hspace*{5mm}  
地虎: \parbox{1.2cm}{  
 張德培\\ 里歐斯}
```

圖 8.2: `\parbox` 指令

其中, `c` 為內定之選項, 可以省略不加。

圖 8.2 是使用段落方塊指令排版的一個例子。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 `\\[4mm]`, 其作用是強迫換行, 並加大行距。第 2 行之基線位置是在「天龍」、「地虎」等文字之下沿。兩個段落方塊都沒有選擇對齊點, 因此 \TeX 選用內定值, 以段落方塊之中央點對齊「天龍」之基線, 也就是其下沿。

段落方塊與迷你版面須設定寬度。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。本例中, 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。本例兩個段落方塊之寬度都是 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。

段落方塊與迷你版面內亦區分段落, 以 `\par` 指令或者空一行區隔段落。但是, 新段落的頭一行不會內縮 (`indent`), 這是和一般文字段落不同之處。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
```

每一段落開頭將內縮 0.6 公分。

圖 8.3 是以 `minipage` 指令環境排版的例子, 其中對齊指令之選項為 `t`。在第 1 行文字之後, 行距特別加大 0.4 公分, 而迷你版面上之沿即對齊第 2 行之基線。本例之第 2 行雖然並無其他文字, 但仍存在一基線。

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` 指令環境內, 其寬度設定為正常行長的 0.85, 並靠左編排。請注意, 在迷你版面內段落開頭文字並不內縮。我們所使用之指令如下:

```
\begin{flushleft}  
\begin{minipage}{.85\textwidth}
```

代表隊員名單如下:

隊長	隊員
陳一軍	山普拉斯 阿格西

```
代表隊員名單如下:\[4mm
\begin{minipage}[t]{1.2cm}
{\ctxfbb 隊長}\ 陳一軍
\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

再舉另一個迷你版面的應用例子。

```
...
\end{minipage}
\end{flussleft}
```

除以上所介紹的之外,段落方塊與迷你版面指令環境還有另外兩個選項。第一個選項用於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,完整的段落方塊指令為:

```
\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:

<code>\width</code>	文字方塊的寬度,
<code>\height</code>	文字方塊頂端到基線之距離,
<code>\depth</code>	文字方塊的深度 (基線以下之長度),
<code>\totalheight</code>	<code>\height</code> 加上 <code>\depth</code> 。

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。

在段落方塊或迷你版面內，若下指令改變行距或選用特定英文字體，其作用只在指令環境內有效。因此，若一開始即選用 `sans serif` 字體，或者特定之中文字體，離開指令環境之後將自動恢復原先之字體。

8.5.2 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 有三種方塊的概念：

- LR 方塊：文字只能由左至右編排，
- 段落方塊：由數行文字由上而下垂直疊起來的方塊；
- 線條方塊 (rule boxes)：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，因為它是由數行文字垂直堆積而成。

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 `\fbbox`，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}\[2pt]</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 stretch)，讓文字均勻分佈於方塊內。若再加入 `\hfill` 或 `\dotfill` 指令，即可設計出特別的文字分佈。例如：

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

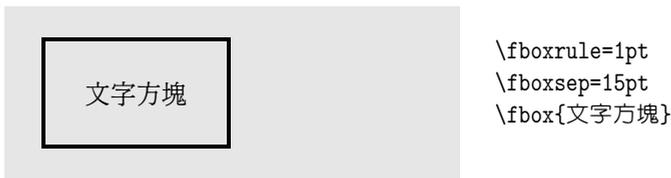
LR 方塊之 `width` 選項除了直接選定長度之外 (如本例之 4cm)，還可以使用下列之單位：`\width`，`\height`，`\depth`，及 `\totalheight`，請參見上一小節之說明。

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。如果要自動換行，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， \TeX 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 LR 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：



`\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

線條方塊指的就是一條直線本身，指令如下：

```
\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，`\rule{1cm}{0.1cm}` 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，`\rule{2cm}{0pt}` 將產生 2 公分的水平空白；反之，`\rule{0cm}{1cm}` 將產生 1 公分之垂直空白。利用以上之指令，

```
\framebox{\rule{3mm}{0pt}\rule{0pt}{3mm}}
```

產生一 3mm 之正方形：。

8.5.4 儲存方塊

若一組文字或符號在文稿中重覆出現, 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 `\newsavebox`; 儲存方塊可以使用 `\sbox` 或者 `\savebox` 指令。舉例言之, 若上一小節之正方形取名為 `\sqe`, 指令為:

```
\newsavebox{\sqe}
\sbox{\sqe}{\framebox{\rule{3mm}{0pt}\rule{0pt}{3mm}}}
```

利用以上之定義, 我們若使用 `\usebox` 指令三次:

```
\usebox{\sqe} \usebox{\sqe} \usebox{\sqe}
```

即可排版出三個連續方塊: 

若使用 `\savebox` 指令儲存方塊, 指令為

```
\savebox{\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 會自動將一部分文字移至下一頁。

註解內之文字將以較小之字體編排, 行距也對應縮小。但中文排版若行距太小, 閱讀不易。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, 如果把全文之行距加大為原來的 1.15 倍, 可在全文設定區加上 `\linespread{1.15}` 指

令。但因為註解之文字較小，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}
```



註解行距

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

\LaTeX 會自動為註解編上阿拉伯數字號碼。在 `book` 或 `report` 文稿類別下，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`。 \LaTeX 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。如果註解是要以 `a`, `b`, `c`, ... 等英文字母編號，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內或表格內加上註解, 我們可以使用下列兩道指令間接排版:

```
\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。

\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`, 註解內容將置於一寬約 1.9 公分的迷你版面內, 排版於版面之兩旁。因為迷你版面寬度甚小, 版面右沿不易對齊。因此, 我們通常加入 `\raggedright` 指令。舉例來說, 下一個段落之邊註的排版指令如下:

```
... 設定右沿不須切齊。 \marginpar{\raggedright  
邊註之內容將排版於正文方塊旁。}
```

邊註之內容將排版於正文方塊旁。

本例中，我們在邊註指令內加上 `\raggedright` 指令，設定右沿不須切齊。邊註將排版於 `\marginpar` 指令出現的那一行上。

邊註之寬度有內定值，但可以變更。若要将寬度設定為 2.0 公分，指令如下：

```
\marginparwidth=2.0cm
```

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 多欄位版面

雜誌常以多欄位版面排版。在 `\documentclass` 文稿類別指令內加入 `twocolumn` 選項，文稿即以兩欄格式排版。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。

有鑑於以上之使用限制，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，可排版 2–10 個欄位之版面，使用上之彈性也較大。不過，`multicol` 巨集套件也有其限制，譬如，在某一欄內下指令引入外製圖形，圖形會排版於下一欄。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位之文字。

欲使用 `multicol` 巨集套件排版多欄位版面，首先須在全文設定區引入巨集套件，接下來即可使用 `multicols` 指令環境排版。指令環境之格式如下：

```
\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。

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。

```
\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?
若從大街小巷充斥著賣名牌的商店來看,
...
這表示民衆對休閒的需求日增。
\end{multicols}
```

圖 8.4: 多欄式版面

圖 8.4 是一個排版兩欄版面之例子。在 `multicols` 指令環境之前, 我們加入 `\raggedcolumns`, 其作用是讓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, 內定值爲 10pt。其次, \LaTeX 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, 內定值爲 12pt, 並有一些伸縮彈性, 必要時可更改之。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 \LaTeX 文稿類別指令內之 `twocolumn` 選項以排版兩欄式版面, 則結束本欄排版之指令爲 `\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。

8.8 comment 巨集套件

文稿寫作經常須反覆修改。修改過程中,我們可以使用 `%` 指令暫時註銷某一段落。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。請注意,以上兩道指令都必須自成一,前端不得有空格,指令末端也不得加入其他文字或指令。我們也可以使用下列指令:

```
\comment ... \endcomment
```

效果相同。

8.9 照列原文

本書列舉許多例子說明 \TeX 的排版指令。要自行以指令照列排版指令事實上並不容易,譬如,排版數學符號 β 之指令為 `$$\beta$`,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 標識符號之指令為 `\LaTeX`,但鍵入指令時,標識符號將直接排版出來,無法看到原始指令。

要將排版指令原文列出,可使用 \TeX 所提供之 `verbatim` 指令環境,或者 `\verb` 指令。英文 `verbatim` 的意義是照列原文的意思。此外,我們也可以使用 Timothy van Zandt 之 `fancyvrb` 巨集套件內之 `Verbatim` 指令環境。以下首先說明 `verbatim` 指令環境及 `\verb` 指令之用法。

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,以下之文字段落內含有 `\LaTeX` 指令,若是一般之文字段落,此一指令將排版為 \TeX 。但因為文字段落置於 `verbatim` 指令環境內,因此指令將照原樣列出。

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，則空格將以 `_` 符號出現。若不特別設定，中文將以明體字排出，本例以 `\ctxfdef` 字體指令選用中文圓體字。

如果照列之文字不超過一行，比較簡單的作法是使用 `\verb` 指令：

`{\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，`\verb` 指令之分隔符號須改用其他字元，如 `-` 號或 `#` 號。

`\verb` 指令之使用有一些限制，例如不能使用於 `\footnote` 指令之內。不過，我們可以使用 `fancyvrb` 巨集套件之功能排除此一限制。使用方法很簡單，引入巨集套件之後，在全文設定區加入下列宣告指令即可：

```
\usepackage{fancyvrb}  
\VerbatimFootnotes
```

`fancyvrb` 巨集套件另外一個功能是簡化 `\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 `ETEX` 標識指令，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，上一行指令須更改為：

```
\DefineShortVerb{\#}
```

8.9.1 Verbatim 指令環境

除了排除註解內使用 `\verb` 之限制外，`fancyvrb` 還提供 `Verbatim` 指令環境。其作用類似 `verbatim`，但功能遠較後者為強。譬如，我們可以設定在每一行照列文

- 1 欲在 `\LaTeX{}` 內照列中文,
- 2 可使用 `Verbatim` 指令環境。

```

\usepackage{fancyvrb}
\ctxfdef{Verbatim}{\ctxfr}
...
\begin{Verbatim}[numbers=left,%
  numbersep=4pt,commandchars=*\#\]
欲在 \LaTeX{ } 內照列中文,
可使用 Verbatim 指令環境。
\end{Verbatim}

```

圖 8.5: `Verbatim` 指令環境

字之前編上號碼,也可以為照列之文字段落加上外框。欲使用 `Verbatim` 指令環境,我們須簡單說明其運作之概念。 \TeX 之指令大都是以反斜線起頭,並以左右大括號界定範圍。因此,如果我們以其他符號替代這三個符號之功能,即可直接照列 `\LaTeX{}`。換言之,經過替代之後,反斜線與左右大括號與一般字元無異,可直接排版出來。

使用 `Verbatim` 指令環境時,我們須經由 `commandchars` 選項選擇三個替代符號。底下的例子中,我們以 `+` 替代原來的反斜線 `\`; 以 `\[` 替代原有之大括號 `{`; 以 `\]` 替代原來之右大括號 `}`。經過以上之替代,原先之反斜線與左右大括號已失去原始之功能,故輸入 `\textit{test}`,即直接原文照列。反之,如果要排版英文斜體字 *test*,我們須輸入: `+textit[test]`。

- 欲在 `\LaTeX{}` 內照列中文,
可使用 `Verbatim` 指令環境。

```

\usepackage{fancyvrb}
\ctxfdef{Verbatim}{\ctxfr}
...
\begin{Verbatim}[commandchars=+\[\]]
欲在 \LaTeX{ } 內照列中文,
可使用 Verbatim 指令環境。
\end{Verbatim}

```

以上的例子以 `+`, `\[`, 及 `\]`, 替代原有之指令符號。若 `Verbatim` 指令環境內之文字內容含有以上 3 個符號,則替代符號須選用其他字元。例如,我們可以星號 `*` 替代反斜線,驚嘆號 `!` 替代左大括號,井字號 `\#` 替代右大括號。

`Verbatim` 巨集套件提供許多控制選項,例如, `numbers=left` 可在每一行文字之前編上號碼, `numbersep=4pt` 則控制編號與文字之間距。本書第 2 章的例子都是以 `Verbatim` 指令環境列出原檔案;每一行指令都加入編號,以方便說明。圖 8.5 是一個加入編號的簡化例子。

9 數學式子

$\text{T}_\text{E}\text{X}$ 以排版數學文稿著稱。Knuth 當初就是為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。 $\text{E}\text{T}_\text{E}\text{X}$ 是由 $\text{T}_\text{E}\text{X}$ 衍生出來的，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， $\text{E}\text{T}_\text{E}\text{X}$ 都能夠自動處理。

若數式相當複雜，或者文稿中用到一些 $\text{E}\text{T}_\text{E}\text{X}$ 所沒有的符號，我們還可使用美國數學學會所開發的 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-E}\text{T}_\text{E}\text{X}$ 巨集套件。本章主要介紹 $\text{E}\text{T}_\text{E}\text{X}$ 的指令，但也將簡單說明 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-E}\text{T}_\text{E}\text{X}$ 之功能。欲深入了解數學排版之各種巨集套件，請參考 [Grätzer \(2000\)](#) 或者 [Mittelbach and Goossens \(2004\)](#)。

9.1 科技文稿之排版規範

$\text{E}\text{T}_\text{E}\text{X}$ 排版數學文稿的能力甚強，但這並不保證我們所排版的文稿一定能達到專業水準。追求優美排版的起點是遵守專業排版之規範。6.4 節會介紹一般性的排版規範，本節進一步說明排版科技文稿的規範及原則，內容主要參考 [Goossens et al. \(1997\)](#), 6.1 節。

排版任何文稿首須注意一致性 (consistency)，科技文稿更不例外。以數學文稿為例，大部分的數學符號須使用數學斜體字。若使用 $\text{E}\text{T}_\text{E}\text{X}$ ，排版時我們只要將符號置於數學模式中， $\text{E}\text{T}_\text{E}\text{X}$ 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、甚至是數字之排版，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，大部分的符號是以

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:

- 單位符號,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 數式環境

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*), 或稱數式環境 (*mathematical environment*)。

隨文數式之數學模式很容易定義,只須在輸入數式時前後加上 \$ 符號即可,排版時 \LaTeX 會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 $\alpha^3 = 8$ 。

如果數式前後忘了加上 \$ 指令,排版時,latex 將發出現錯誤訊息,因為 α 指令只能用於數學模式內。反之,如果我們排版的式子是: 若 $a = 2$ 則 $2a = 4$; 而輸入時忘了加上 \$ 指令,排版時不會出現錯誤訊息,但結果變成: 若 $a=2$ 則 $2a=4$ 。

欲進入數學模式,除了 \$... \$ 指令外,另外一種指令格式是:

$\backslash\text{begin}\{\text{math}\} \dots \backslash\text{end}\{\text{math}\}$

此外，我們也可以用 $\backslash(\dots \backslash)$ 。顯然，第一種指令型式最簡單，因此也最常使用。排版展示數式也須進入數式環境。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，每一行數學式都會自動編上號碼。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。我們還可以使用另一個簡化之指令型式：`$$... $$`。

若使用第3種簡化指令型式，排版上面展示數式之指令如下：

```
\[
\phi = \sum^{\infty}_{t=0} \beta^t U(c_t, x_t) \mbox{.}
\]
```

請注意行末中文句點之輸入方式。因為中文句點並非 \LaTeX 之數學符號，故須置於 `\mbox` 指令內。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，如上例所示；但也可以直接加在數式的前後。

較複雜的數學式可能橫跨兩行以上，例如：

$$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t,$$
$$\varepsilon_t = \rho \varepsilon_{t-1} + u_t.$$

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 (頁 60) 曾說明中文稿之輸入方法，其中較重要的兩點是，換行應在標點符號之後，以及中文字間不應有空白。輸入數學文稿時，另請注意以下三點：

- 隨文數式前後請留一空格。“若 α 之值等於 2” 是不對的，排版之後會變成「若 α 之值等於 2」。正確的輸入法為“若 α 之值等於 2”；排版結果為「若 α 之值等於 2」。
- 展示數式上下不須多留一空白， \LaTeX 會自行調整間距。
- 不管是隨文數式或展示數式都應加上標點符號。隨文數式列於行文當中，很自然會加上標點，但很多人在輸入展示數式時會忽略標點符號。



輸入數學式

\LaTeX 排版數學式子的能力很強，但輸入文稿時若不注意細節，不可能得到美好的結果。

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 \LaTeX 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 $	$\$ a < b+c = d \$ \backslash$
$y'' = f\{y', y(x)\}$	$\$ y'' = f\{y', y(x)\} \$$

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 。上下標若只有單一符號或數字，大括號可以省略。因此，上例亦可輸入為 $\$a_n\$$ 。若同時有上下標時，輸入上下標之順序並不重要。因此，鍵入 $\$x^2_t\$$ 或者 $\$x_t^2\$$ 均可產生 x_t^2 。

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}	$\$_{y^2}\$$	$A_{i,j,k}^{-n/2}$	$\$A_{i,j,k}^{-n/2}\$$
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 $\frac{}{} \text{\LaTeX}$ 為之。例如：

$a/(m+n)^2$ 乘上 $1/(x+y)$ 等於	$\$a/(m+n)^2\$$ 乘上 $\$1/(x+y)\$$ 等於
$\frac{a}{(m+n)^2(x+y)}$	$\$[\frac{a}{(m+n)^2(x+y)}\text{\LaTeX}]\$$

分式指令 `\frac` 包含兩個部分，第一部分為分子，第二部分為分母；兩項都須以大括號括起來。本例亦顯示，展示數式指令之前後不須空一行， \TeX 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：

$\frac{x+y}{1+\frac{y}{x+y}}$	<pre>\[\frac{x+y}{1 + \frac{y}{x+y}} \]</pre>
-------------------------------	--

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：

$u = \sqrt[3]{-q + \sqrt{q^2 + p^3}}$	<pre>\[u = \sqrt[3]{-q+\sqrt{q^2+p^3}} \]</pre>
---------------------------------------	--

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，而且必須置於數式環境中。本節將依序介紹各種數學符號之指令。本章僅介紹數學符號，如果你想知道 \TeX 到底有那些符號可用，請見 [Pakin \(2005\)](#)。

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`。底下是另一個例子：

表 9.1: 希臘字母符號

α	<code>\alpha</code>	θ	<code>\theta</code>	\omicron	<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>		

$$(\gamma^\mu - m)\psi = 0.$$

`\[(\gamma^\mu - m)\psi = 0.\]`

在數學模式中， \LaTeX 會自行調整各符號的間距。因此，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。輸入上述指令時，若將空格全部去除，變成：`\[(\gamma^\mu-m)\psi=0.\]`，排版結果相同。

除了希臘字母之外， \LaTeX 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`。

9.3.2 粗體數學符號

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 \LaTeX 中，選用數學字體之指令包括：

`\mathrm` `\mathtt` `\mathbf`
`\mathsf` `\mathit` `\mathcal`

不過，數學字體指令只能改變英文字母、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：

$$\mathbf{A} = \mathbf{B}^0(x)$$

`\[\mathbf{A}=\mathbf{B}^0(x)\]`

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>			

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,

$$\alpha/\beta \neq \alpha/\beta.$$

```
\usepackage{bm}
...
\[ \bm{\alpha/\beta} \neq \alpha/\beta. \]
```

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}` 可排出 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。

$$\mathbf{A} = \mathbf{B}^0(x)$$

$$A = B^0(x)$$

```
\boldmath
\[ \mathbf{A}=\mathbf{B}^0(x) \]
\unboldmath
\[ A=B^0(x) \]
```

有些數學符號並無粗體字型, 因此時即用了數學粗體字指令, \LaTeX 仍以普通字體排版。

9.3.3 函數符號

依據排版規範, 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 \LaTeX 中, 函數應以指令輸入: `$\log xy$` , 排版結果為 $\log xy$ 。表 9.2 列出常用之函數符號指令。

統計學中經常使用數學期望值 (expectation) 與變異數 (variance) 符號。函數符號應使用正體字, 因此應排成 $E(x)$ 與 $\text{var}(x)$ 。但這兩個函數並無現成的指

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>	\bigodot	<code>\bigodot</code>	\bigotimes	<code>\bigotimes</code>	\bigoplus	<code>\bigoplus</code>
\biguplus	<code>\biguplus</code>	\bigsqcup	<code>\bigsqcup</code>				

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, 以方便輸入, 並避免錯誤。巨集指令之定義方法, 請見第 14 章。

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`, 加總函數符號指令為 `\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:

$$\int_0^1 f(x)dx \neq \sum_{i=1}^n x_i$$

```
\[
\int^1_0 f(x)dx \neq
\sum^n_{i=1} x_i
\]
```

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, 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_{t=0}` 可排版出: $\sum_{t=0}^t$ 。不過, 使用 `\limits` 指令時, 上下行距會自動加大一些, 版面整體效果並不好。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。

以上 `\limits` 指令的一個應用是排版統計學或計量經濟學的迴歸方程式 (regression equation)。迴歸方程式之估計值的下方常須列出標準差。L^AT_EX 並無現成之指令排版此種數學式, 故我們須以一般指令指令排版。圖 9.1 的例子中, 我們先在全文設定區定義 `\tb` 指令, 再以之排版迴歸式。請注意, `\tb` 巨集指令必須在數式環境內才能使用。

$$y = 1.23 x_1 + 2.34 x_2 + \epsilon.$$

(2.23) (1.22)

```
\def\tb#1#2{\mathop{#1\phantom{\sum}}\limits_{\displaystyle #2}}
...
\[
y = \tb{1.23}{(2.23)}x_1 + \tb{2.34}{(1.22)}x_2 + \epsilon.
\]
```

圖 9.1: 迴歸式

使用 `\tb` 指令時必須填入兩項參數, 第一項為迴歸式之估計值, 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)。上面說明, `\limits` 指令之功能是把下標置於積分或加總符號之正下方, `\tb` 巨集指令即利用此一性質, 將 (2.23) 排版於 1.23 之下。巨集指令之定義中使用了 ``, 其作用是以 \sum 符號之高度作為以上兩個數字之間距。若要把間距加大一些, 我們須代入較大的符號。譬如, 以 `\big(` 或者 `\Big(` 替代定義式中之 `\sum`。

9.3.5 箭頭符號

接下來的幾個小節內, 我們將分間別類介紹一些常用的數學符號。這些符號依其性質可分為: 箭頭符號 (arrow symbols), 相對關係符號 (relation symbols), 與二元運算符號 (binary operators)。

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, 用於排版流程圖等。其中, \leftarrow 符號之指令為 `\leftarrow`, 但也可以使用較簡單的 `\gets` 指令。同樣的, \rightarrow 右箭頭符號之指令為 `\rightarrow`, 但可以使用較簡化的 `\to` 指令。

數學之向量符號與箭頭符號之樣子很像, 但位置高低不同。要排版向量符號, 應使用 `\overrightarrow` 指令:

\vec{a} $\overrightarrow{a+b}$ $\overrightarrow{a+b+c}$	<pre>\\$\\overrightarrow{a}\$ \\[3pt] \\overrightarrow{a+b}\$ \\[3pt] \\overrightarrow{a+b+c}\$</pre>
---	---

反之, `\overleftarrow{x+y}` 指令可排出 $\overleftarrow{x+y}$ 。以上之向量符號的長度會隨著符號的寬度自行調整。L^AT_EX 還有其他可自行調整大小之符號, 譬如左右中括號。

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>	\Updownarrow	<code>\Updownarrow</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>
\lefttharpoonup	<code>\lefttharpoonup</code>	\righttharpoonup	<code>\righttharpoonup</code>	\nwarrow	<code>\nwarrow</code>
\rightrightarrows	<code>\rightrightarrows</code>	\leadsto	<code>\leadsto</code>		

這在排版大型矩陣時很方便, 請見 9.4 節 (頁 148)。

雖然表 9.4 已提供各式各樣的箭號, 但是再多的符號似乎也無法滿足所有人的需求。為了應付數學文稿排版的需求, 美國數學學會 (American Mathematical Society) 又請專家設計一些符號, 並提供 `amssymb` 巨集套件以方便引用。譬如, 數學文稿中有時候使用 \because 符號代表「因為」, 而 \therefore 符號則代表「所以」, 但 \TeX 並沒有這兩個符號。`amssymb` 巨集套件則提供這兩個符號。在全文設定區引入 `amssymb` 巨集套件:

```
\usepackage{amssymb}
```

文稿內即可使用 `\because` 與 `\therefore` 指令排版這兩個符號。

巨集套件 `amssymb` 所提供的符號可簡單區分為 5 大類, 以下將依序介紹。表 9.5 列出 AMS 之箭頭符號。

9.3.6 相對關係符號

相對關係符號是指諸如大於或小於之符號。 \TeX 之相對關係符號列於表 9.6, 其中 `\leq` 指令用以排版 \leq , 但指令可簡化為 `\le`。同理, `\geq` 指令也可簡化為 `\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) 排版結果稍有不同; 排版專家認為後者較佳。

表 9.6 所列之相對關係符號絕大部分是 \TeX 系統所提供的。但是, 其中的 \sqsubset 與 \sqsupset 兩個符號則是 \TeX 特別提供的, 指令分別為 `\sqsubset` 與 `\sqsupset`。欲

表 9.5: AMS 箭頭符號 (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>
\Rrightarrow	<code>\Rrightarrow</code>	\Rleftarrow	<code>\Rleftarrow</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>
\upuparrows	<code>\upuparrows</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>	\nleftarrow	<code>\nleftarrow</code>	\nLeftarrow	<code>\nLeftarrow</code>

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, 而文稿中使用這兩個符號指令, 排版時會出現指令未會定義 (undefined) 之錯誤訊息。

表 9.6 中雖然有大於與小於符號, 但有些數學文稿須使用大於與小於並列之符號。amssymb 巨集套件提供之相對關係符號即包含 \lesseqgtr , 指令是 `\lesseqgtr`。如果是大於, 等於與小於並列, 可使用 `\gtreqless`: \gtrless , 或者 `\gtreqqlless`: \gtrqlless 。相反的, 輸入 `\lesseqgtr` 指令可排出 \lesseqgtr , 而 `\lesseqqgtr` 指令則排出 \lesseqqgtr 。表 9.7 列出 AMS 之雙元關係符號。

除了雙元關係符號之外, AMS 另有 negated binary symbols, 我們稱之為負雙

表 9.6: 相對關係 (relation) 符號

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
$<$	<code>\prec</code>	$>$	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>	$ $	<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>				

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>	\approxeq	<code>\approxeq</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>	\lesseqqgtr	<code>\lesseqqgtr</code>
\gtreqqless	<code>\gtreqqless</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>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</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>	\prec	<code>\prec</code>	\succsim	<code>\succsim</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\vartriangleright	<code>\vartriangleright</code>	\parallel	<code>\shortparallel</code>	\Vdash	<code>\Vdash</code>
\vartriangleleft	<code>\vartriangleleft</code>	\smile	<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	\leq	\leqq
\ngtr	\ngeq	\ngeqslant
\ngeqq	\geq	\geqq
\lvertneqq	\gvertneqq	\lesim
\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	\subseteq
\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。此表內之符號有很多是表 9.7 內符號之否定。譬如, `\leqq` 指令產生 \leq 符號; 而 `\nleqq` 則產生 \nleq 。

9.3.7 雙元運算符號

以上兩小節分別介紹箭頭符號與相對關係符號, 接下來表 9.9 列出雙元運算符號。常用的雙元關係符號如 `\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 \TeX 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`。加號與等號都太小, 間距也不正確。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`。

在數學模式內, `\dagger` 指令可排版 \dagger 符號; `\ddagger` 指令可排版 \ddagger 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 \TeX 的雙元運算符號外, AMS 又提供更多的符號, 如表 9.10 所示。其中, `\centerdot`

表 9.9: 雙元運算符號

±	<code>\pm</code>	∩	<code>\cap</code>	◇	<code>\diamond</code>	⊕	<code>\oplus</code>
∓	<code>\mp</code>	∪	<code>\cup</code>	△	<code>\bigtriangleup</code>	⊖	<code>\ominus</code>
×	<code>\times</code>	⊕	<code>\uplus</code>	▽	<code>\bigtriangledown</code>	⊗	<code>\otimes</code>
÷	<code>\div</code>	∩	<code>\sqcap</code>	◁	<code>\triangleleft</code>	⊘	<code>\oslash</code>
*	<code>\ast</code>	∩	<code>\sqcup</code>	▷	<code>\triangleright</code>	⊙	<code>\odot</code>
★	<code>\star</code>	∨	<code>\vee</code>	◁	<code>\lhd</code>	○	<code>\bigcirc</code>
○	<code>\circ</code>	∧	<code>\wedge</code>	▷	<code>\rhd</code>	†	<code>\dagger</code>
●	<code>\bullet</code>	\	<code>\setminus</code>	◁	<code>\unlhd</code>	‡	<code>\ddagger</code>
·	<code>\cdot</code>	∩	<code>\wr</code>	▷	<code>\unrhd</code>	⊚	<code>\amalg</code>

表 9.10: AMS 雙元運算符號 (amssymb 巨集套件)

‡	<code>\dotplus</code>	⋈	<code>\ltimes</code>	∖	<code>\smallsetminus</code>
⋈	<code>\rtimes</code>	⊎	<code>\Cap, \doublecap</code>	⋈	<code>\leftthreetimes</code>
⊎	<code>\Cup, \doublecup</code>	⋈	<code>\rightthreetimes</code>	⌒	<code>\barwedge</code>
⌒	<code>\curlywedge</code>	∨	<code>\veebar</code>	∨	<code>\curlyvee</code>
⌒	<code>\doublebarwedge</code>	⊖	<code>\boxminus</code>	⊖	<code>\circleddash</code>
⊖	<code>\boxtimes</code>	⊗	<code>\circledast</code>	⊖	<code>\boxdot</code>
⊙	<code>\circledcirc</code>	⊕	<code>\boxplus</code>	·	<code>\centerdot</code>
∗	<code>\divideontimes</code>	⊔	<code>\intercal</code>		

比 \TeX 之 `\cdot` 要稍大一些。欲使用這些符號，我們必須先引入 `amssymb` 巨集套件。

除了以上各表所列的符號之外， \TeX 還有一些不易歸類的符號。有些用於數學與自然科學中，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 \TeX 的 `latexsym` 巨集套件所提供：

□ ◇ ∪ ∋ ◁ ▷ ≤ ≥ ∼

連同前面所介紹的 `□` 與 `□`，`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。不過，此一套件所定義之符號也包含在 `amssymb` 巨集套件之內，故直接引入後者也可以使用這些符號。

表 9.11 列有幾個樂譜符號，如 `♯` 與 `b` 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 \TeX 系統為基礎的樂譜排版系統；有興趣深入了解者，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` 可排版一正方形：□。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 □。另外，`\blacksquare` 可排版一實心小正方形：■，有人用於標示數學證明結束。

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>\bot</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 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>
∇	<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>\circledS</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>

最後，數學中之 \bmod 函數有兩種指令型式，第一種稱為二元型式 (binary)，輸入指令為 `\bmod`；第二種稱為括號型式 (parenthesized)，指令為 `\pmod`。譬如，輸入 `a\bmod b` 指令結果為 $a \bmod b$ ；反之，輸入 `\pmod{a+b}` 指令則排出： $(\bmod a + b)$ 。

如果只是偶而排版數學式，要記住以上所有的符號指令其實不可能。如果你使用 WinEdt 軟體，其數學符號圖形之功能對於排版數學式很有幫助。點選 WinEdt 視窗上之 \sum 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 \sum 圖像，即關閉工作列。

9.3.8 數學重音符號、底線與上線

第 5 章的表 5.1 (頁 56) 曾介紹重音符號 (accents)，例如，在一般文稿內，輸入 `\=o`，

表 9.13: 數學重音符號

\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>
\breve{o}	<code>\breve{o}</code>	\tilde{o}	<code>\tilde{o}</code>				

可排版 \bar{o} 。以上所介紹的數學符號也可附加上重音符號。例如, $\hat{\beta}$ 或者 \bar{a} 。表 9.13 列出所有的數學重音符號。有些數學重音符號可以放大, 例如, `\hat` 符號若要加大一些, 可使用 `\widehat`。因此, 欲排版 $\widehat{x + y/\widehat{xy}}$, 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,

```
 $i+j \neq \vec{\imath} + \bar{\jmath}$ 
```

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 3x.
```

```
value is \underline{3x}.
```

若要加入上線, 我們可以使用 `\overline` 指令。例如:

```
 $\overline{\overline{x^2 + 1}}$ 
```

```
\overline{\overline{x^2+1}} \\[4pt]
\overline{\overline{x^2+1}}
```

`\overline` 與 `\underline` 的長度都會隨著數學變數之大小而自動調整。

上線與底線是指加在符號上方或底下加一直線。有時候, 我們須加入頂大括號或底大括號。頂括號之指令為 `\overbrace`, 底括號為 `\underbrace`。例如:

```
 $\overbrace{a + \underbrace{b + c}}$ 
```

```
\overbrace{a+\underbrace{b+c}}
```

以上是隨文數式例子。展示數式中，頂括號與底括號若再加上上下標，排版結果如下所示：

$$\underbrace{a + b + \cdots + x + y + z}_{26}^{\quad 24}$$

```
\[
\underbrace{a + \overbrace{b
+\cdots+x+y}^{24}+z}_{26}
\]
```

`\overbrace` 與 `\underbrace` 一樣，其大小也是隨著其所涵蓋的數學式之大小而自動調整。

9.3.9 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：

$$A \stackrel{a}{\rightarrow} B$$

$$\vec{x} \stackrel{\text{def}}{=} (x_1, \dots, x_n)$$

```
$A \stackrel{a}{\rightarrow} B$ \\[4pt]
$\vec{x} \stackrel{\mbox{def}}{=}
(x_1, \ldots , x_n)$
```

`\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 TeX 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：

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$$

```
\[
{n+1 \choose k} =
{n \choose k} + {n \choose k-1}
\]
```

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，

$$\sum_{\substack{0 \leq i \leq n \\ 0 \leq t \leq T}} U(c_{it}, x_{it})$$

```
\[
\sum_{\scriptstyle 0 \leq i \leq n \atop \scriptstyle 0 \leq t \leq T}
U(c_{it}, x_{it})
\]
```

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 將以內定之字體編排，結果會顯得太小。

9.3.10 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，可算出 $X = x_1 + x_2 + \dots$ 。

把 `x_1, x_2, \dots` 各項相加，可算出 `$X=x_1+x_2+ \cdots$`。

本例中，`\cdots` 所產生的連續點，其位置稍高於 `\ldots`。如上例所示，`\cdots` 通常是用於 $+$ ， $-$ ，與 $=$ 號之間。相對的，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`

垂直與對角方向之連續點可用於排版矩陣與行列式，下一節將舉例說明其用法。請注意，`\cdots`，`\vdots`，與 `\ddots` 只能用於數式指令環境中；但是 `\ldots` 可以用於數式環境，也可以用於一般的文字排版。

9.4 矩陣與行列式

矩陣的特徵是多項符號規則性的排列，行列式與矩陣類似，但無左右括號。矩陣與行列式通常使用 `array` 指令環境排版，請見底下的例子：

$$\begin{array}{ccc} a + b + c & m + n & xy \\ a + b & p + n & yz \\ b + c & mn & xyz \end{array}$$

```
\[ \begin{array}{clr} \backslash
a+b+c & m+n & xy \backslash
a+b & p+n & yz \backslash
b+c & mn & xyz
\end{array} \]
```

`array` 指令環境只能用於數學模式內。橫行各項之間是以 `&` 區隔，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加上換行指令。使用 `array` 指令環境時，我們須加上選項以控制每一直欄內各數字或符號要居中排列、靠左或

靠右。本例爲了說明起見，使用 {clr} 選項，因此第一欄文字居中 (center)，第二欄文字靠左 (left)，第三欄文字靠右 (right)。

如果要排版矩陣呢？很簡單，只要在行列式兩旁加上大括號即可。若爲方括號，指令分別爲 \left[與 \right]。方括號之大小會因應數學表列大小自行調整。

$A = \begin{bmatrix} a+b & mn & xy \\ a+b & pn & yz \\ b+c & mp & xyz \end{bmatrix}$	<pre>\[A = \left[\begin{array}{c} \begin{array}{ccc} a+b & mn & xy \\ a+b & pn & yz \\ b+c & mp & xyz \end{array} \end{array} \right]</pre>
--	---

版面上的每一行文字都座落在各自的基線 (baseline) 上。排版行列式時，全表之中央將對準基線。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：

$\Delta = \begin{pmatrix} \alpha + b & \phi \\ \gamma & m\beta \end{pmatrix}$	<pre>\[\Delta = \left(\begin{array}{c} \begin{array}{cc} \alpha+b & \phi \\ \gamma & m\beta \end{array} \end{array} \right)</pre>
---	---

9.4.1 界限符號

以上排版矩陣與行列式之左右圓括號與方括號又稱爲界限符號 (delimiter)；其特點是，我們可以加上指令以放大之。界限符號指令可以直接用於排版簡單的數學式，如 (a/b)。但是，當分子與分母變大時，除號與圓括號都應該加大，譬如，(a²/b³)。欲放大符號，我們可以在指令之前加上 \big 指令。故上式之排版指令爲： $\big(a^2\big/\big(b^3\big)$ 。表 9.14 列出全部的界限符號。

放大界限符號之指令事實上有好幾個，譬如，\Big(所產生的符號即比 \big(大一些。以右圓括號爲例，全部的調整大小之指令如下：

$$):) \quad \big):) \quad \Big):) \quad \bigg):) \quad \Bigg):)$$

表 9.14: 界限符號 (delimiter)

(({	\{	⌊	\lfloor	↑	\uparrow
))	}	\}	⌋	\rfloor	↓	\downarrow
[[⟨	\langle	↕	\Uparrow
]]		\	⟩	\rangle	↕	\Downarrow
⌈	\lceil	⌉	\rceil	/	/	↕	\updownarrow
				\	\backslash	↕	\Updownarrow

雖然符號可以放大, 但如果分子與分母都相當複雜時, 我們並不易判斷應使用那一級之放大指令。此時, 我們可直接使用 `\left` 與 `\right` 指令讓 \TeX 作調整。例如:

$A = \left(\frac{dx}{ds} \right) / \left(\frac{dy^B}{ds^A} \right)$	<pre>\[A = \Big(\frac{dx}{ds}\Big)\Big/ \Big(\frac{dy^B}{ds^A}\Big) \]</pre>
$A = \left(\frac{dx}{ds} \right) / \left(\frac{dy^B}{ds^A} \right)$	<pre>\[A = \left(\frac{dx}{ds}\right)\bigg/ \left(\frac{dy^B}{ds^A}\right) \]</pre>

第一行是自行加上 `\Big` 指令, 第二行則是利用 `\left` 與 `\right` 指令, 由 \TeX 調整。比較兩者之排版結果可知, 若是自行調整, 我們須使用更大一級的 `\bigg` 指令。顯然, 讓 \TeX 調整大小是比較方便的。不過, 要注意的是 `\left` 與 `\right` 指令須成對出現。若僅輸入 `\left`, 忘了輸入 `\right`, 排版時將出現錯誤訊息。

有些界限符號是左右成對出現的, 如左右括號。為了易為分辨, 成對出現之放大指令可加入 `l` 與 `r`。例如, `\bigl(` 與 `\bigr)`; 又如, `\Bigl[` 與 `\Bigr]`。另外, 大括號之輸入指令為 `\left\{` 與 `\right\}`。

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, 我們必須加上 `\right.` 指令代替右括號, 例如:

$x = \begin{cases} y & \text{if } y > 0, \\ z + y & \text{otherwise.} \end{cases}$	<pre>\[x = \left\{ \begin{array}{ll} y & \mbox{if } y > 0, \\ z+y & \mbox{otherwise.} \end{array} \right. \]</pre>
--	--

本例中, 數學式內用了英文字 “otherwise”。在數學式內排版一般文字時, 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或中文句點時, 也必須置於 `\mbox` 指令內, 例如:



$$x = \begin{cases} y & \text{若 } y > 0, \\ z + y & \text{其他情況。} \end{cases}$$

```
\[ x = \left\{
\begin{array}{ll}
y & \text{若 } y > 0, \\
z+y & \text{其他情況。}
\end{array} \right. \]
```

同理, 如果數學式子只需右括號, 則使用 `\left`. 替代左括號, 例如,

$$\left. \begin{array}{l} \Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt \\ \Gamma(z+1) = z\Gamma(z) \end{array} \right\}$$

```
\[ \left.
\begin{array}{l}
\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt \\
\Gamma(z+1) = z\Gamma(z)
\end{array}
\right\}
```

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: ${}_tX_{t+1}$ 。若 `_t` 之前未加上 `\mbox{}`, 排版時將出現錯誤。

底下是一個較複雜的矩陣, 但只要把段落分清楚, 各項指令並不難了解:

$$\left(\begin{array}{cc|cc} a & b & & \\ c & d & & A \\ & & m & n \\ B & & o & p \end{array} \right)$$

```
\[ \left(
\begin{array}{cc|cc}
a & b & & \\
c & d & & A \\
& & m & n \\
B & & o & p
\end{array}
\right)
```

其中, 垂直線指令為 `|`, 這也是界限符號, 故本例使用 `\left|` 與 `\right|` 指令, 自動調整大小。

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)
```

9.5 定義與定理

數學文稿常須排版定義與定理。 \TeX 原提供 `\newtheorem` 指令，讓使用者自行設計定義與定理的排版格式。1980 年代晚期，美國數學學會商請 Frank Mittelbach 與 Rainer Schöpf 開發 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ 巨集套件，以強化 \TeX 之數學排版的功能。 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ 包含幾套巨集套件，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，後者的主要功能即在於排版定義與定理。

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ 是由美國數學學會商請專家所開發，有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), 在 \TeX 中難以解決，但使用 `amsmath` 可輕易辦到。另外， \TeX 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，本節首先介紹 `amsthm` 巨集套件，其主要功能是排版數學文稿的定理與定義，

9.5.1 `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。圖 9.2 是排版定理的一個例子。如果是排版定義，則「定理」兩字須替換為「定義」。雖然在數學上，定義與定理之性質截然不同，但從排版的角度來看，版面格式其實相當接近。因此，`amsthm` 巨集套件利用同一套指令排版定理與定義。

若是排版定理，我們必須先使用 `amsthm` 之指令創造一指令環境，再利用它來排版文稿內之定理。定義或引理也是以同樣方式處理。以定理為例，創造指令環境時須取一名字，並選擇排版格式。指令環境是以 `\newtheorem` 指令創造；而 `\theoremstyle` 指令則選擇排版格式。`amsthm` 巨集套件提供三種現成之格式：

- `plain`: 格式最醒目，
- `remark`: 格式最不醒目，
- `definition`: 介於兩者之間。

圖 9.2 例子的第 2 行指令指定使用 `plain` 格式，第 3 行指令則以 `\newtheorem` 指令創造一個排版定理之指令環境，取名為 `thm`。接下來，我們即可使用 `thm` 指令環境排版定理。

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。透過仔細之計算可得証, $E(v_{km}^{kn-1}) = E(v_m^{n-1})$ 。 \square

```

\usepackage{amsthm}
\theoremstyle{plain}
\newtheorem{thm}{\ctxfbb 定理}
\renewcommand{\proofname}{\ctxfk 證明}
...
\begin{thm}
若  $v$  均等分佈於  $[a, b]$ ,
則  $E(v_{km}^{kn-1}) = a + (b - a)(n - m)/n$ 
\end{thm}
\begin{proof}
獨立抽取  $N - 1$  個均等分配於  $U[0, 1]$  的隨機變數。
透過仔細之計算可得証,
 $E(v_{km}^{kn-1}) = E(v_m^{n-1})$ 。
\end{proof}

```

圖 9.2: 定理與證明

底下先以英文稿為例, 簡單說明各種格式之差異。若是英文稿, 而且第 2 行指令仍選用 `plain`, 但第 3 行設定指令更改如下:

```
\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。相反的, 若第 2 行指令改用 `remark` 格式, 則定理之標題將以斜體字排版, 變成 *Theorem 1.*。定理陳述本身則是標準的英文正體字。

不同的格式會選用不同的英文字體排版 `Theorem` 英文字, 但是, 英文字體指令對中文無效, 故中文稿須重新定義標題。圖 9.2 例子第 3 行即設定標題為「定理」, 並選用粗黑體排版。本定理雖然使用 `\mbox{E}` 指令讓期望值之符號以標準字體排版, 但排版結果卻是斜體字 E 。其中的原因是, 本例選擇 `plain` 格式, 故定理陳述內定以英文斜體字編排。若把 `\mbox{E}` 指令改為 `\mbox{\textup{E}}`, 結果即為正確。

依同樣的原理, 我們可創造一指令環境排版定義或預備定理 (lemma)。譬如, 在全文設定區輸入下列兩行指令:

```

\theoremstyle{definition}
\newtheorem{def}{\ctxfbb 定義}

```

文稿內即可使用 `def` 指令環境排版定義, 格式則選用 `definition`。

9.5.2 定理編號與排版格式

利用 `\newtheorem` 所定義之指令環境排版, 定義或定理會自動加上編號。如果不要編號, 應改用 `\newtheorem*` 指令。若選擇自動編號, `\newtheorem` 指令另提供兩個選項以控制編號方法。以 `book` 文稿類別排版書籍時, 我們通常使用 `\chapter` 排版章標題。因此全書之定理可能是從頭開始連續編下來, 也可能以章為單位編號。假設是以章為單位, 則第 3 章內的定理應編為 **3.1**, **3.2** 等。若要以章為單位編號, 圖 9.2 第 3 行定義 `thm` 之指令應加上 `chapter` 選項, 變成:

```
\newtheorem{thm}{\ctxfbb 定理}[chapter]
```

數學文稿中有定理、定義、預備定理等。有些作者喜歡將定理與引理混合編號, 譬如, 定理 2, 引理 3, 引理 4, 定理 5 等等。若要以此方式排版, 則創造 `lem` 指令環境時, 須加入 `thm` 選項:

```
\newtheorem{lem}[thm]{\ctxfrb 引理}
```

以上指令定義 `lem` 指令環境, 其中之 `thm` 選項表示新創造之 `lem` 指令環境將與 `thm` 指令環境混合編號。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}
```

正常情況下, 定理編號是排於「定理」二字之後。如果要把編號置於「定理」之前, 應在全文設定區加入 `\swapnumber` 指令。

9.5.3 定理證明

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` 巨集套件提供 `proof` 指令環境以排版證明。使用 `proof` 指令環境時, 證明開始處會自動加上 *Proof* 一字。圖 9.2 之例子為中文稿, 因此我們將 *Proof* 改為中文楷體。如指令第 4 行所示, 設定方法是將 `\proofname` 重新定義為中文字。

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; 證明結束處則加上 \square 符號。後者之符號可以重新設定。例如, 在全文設定區加入下列指令:

```

\newtheoremstyle{thmsty}%取名
  {3pt}%      上方間距
  {3pt}%      下方間距
  {}%         選用定理、定義陳述之英數字體
  {\parindent}% 首行內縮距離
  {\sffamily}% 選用標題之英數字體
  {:}%        標題後加上冒號
  {.5em}%     標題與定理陳述之間距
  {}%         特殊設定,請見巨集指令說明檔。

```

圖 9.3: 設計 amsthm 排版格式

```
\newcommand{\qedsymbol}{\Box}
```

符號將為 □, 比原正方形符號小一些。

如上所述, amsthm 巨集套件提供三種格式: plain, definition 與 remark。如果你不滿意這三種格式, 可自行設計。設計排版格式須使用 \newtheoremstyle 指令。圖 9.3 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 9.2 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, 圖 9.3 所設計格式中, 標題之英數字體以 \sffamily 字體族排版; 但定理陳述內容則以標準字體編排。若定理陳述要改用英文斜體字, 則第 4 行指令應由 {} 改為 {\itshape}。最後, 如果希望「定理」兩字要單獨排為一行, 定理陳述排版於其下, 則倒數第 2 行大括號內應改填入 \newline。

經過以上之定義, 除了原有三種格式之外, 我們也可以選用 thmsty 格式排版定理與定義。欲進一步了解 amsthm 巨集套件之指令, 請參考其說明檔。

9.6 amsmath 巨集套件

前面所舉的數式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 提供 eqnarray 指令環境以排版多行數學式, 但其排版效果並不理想, 原因是等號之兩邊會留下太大的空白。欲排版多行之數學式, 應使用 amsmath 巨集套件所提供之指令環境 align。amsmath 巨集套件除了解決等號左右多出空白的問題之外, 還有其他的功能, 本節也將作簡單介紹。

依排版規範, 多行式子並排時, 若每一行中都有一個等號 (或者大於、小於符號) 時, 等號應上下對齊。若以 amsmath 巨集套件排版多行之數式, 可使用 align 指令環境讓等號上下對齊。

$$\begin{aligned} a_1 &= b_1 + c_1, & (1) \\ a_2 &= c_2 + e_2. \end{aligned}$$

```
\usepackage{amsmath}
...
\begin{align}
a_1 &=& b_1+c_1, \\
a_2 &=& c_2+e_2. \notag
\end{align}
```

首先，於全文設定區引入 `amsmath` 巨集套件。在 `align` 指令環境內，欲上下對齊之符號，其前面須加上 `&`。以 `align` 指令環境排版，各式子會自動編號。欲取消某一式之編號，須在該式末端加上 `\notag` 指令。如果各式子都不編號，應直接使用 `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， \LaTeX 自動加上左右括號；若使用 `\tag*` 指令，使用者須自行填入括號。

$$\begin{aligned} a_1 &= b_1 + c_1, & (3a) \\ a_2 &= c_2 + e_2. & (3b) \end{aligned}$$

```
\usepackage{amsmath}
...
\begin{align}
a_1 &=& b_1+c_1, \tag{3a} \\
a_2 &=& c_2+e_2. \tag*(3b)}
\end{align}
```

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。

$$\begin{aligned} a_1 &= b_1 + c_1 + d_1, & x_1 &= y_1 + z_1, & (1) \\ a_2 &= b_2 + c_2 + d_2, & x_2 &= y_2 + z_2. & (2) \end{aligned}$$

```
\usepackage{amsmath}
...
\begin{align}
a_1 &=& b_1+c_1+d_1, & & x_1 &=& y_1+z_1, \\
a_2 &=& b_2+c_2+d_2, & & x_2 &=& y_2+z_2.
\end{align}
```

有時候，上下兩行或多行排列的數學式應視為一組數式，因此只需一個編號，此時我們可使用 `split` 指令環境。`split` 指令環境可以讓等號上下對齊，但並無自動編號功能。若要加入編號，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。底下是一個例子：

本例是以 `align*` 指令環境編排, 因此數式並不編上號碼。

9.6.1 引述數式

以 `equation` 或 `align` 排版時, 數學式會自動編上號碼。若使用 `book` 文稿類別排版, 編號將以章為單位。例如, 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, 可先以 `\label` 指令加上標籤, 再使用 `\ref` 指令引述。以下的例子中, 數式之編號是以 `\tag` 指令自行加上。本例中, 兩行數學式之標籤分別是 `xy` 與 `ab`:

$z = x + y,$	(2a)	<code>\usepackage{amsmath}</code>
$c = a + b.$	(2b)	<code>...</code>

```
\begin{align}
z &= x+y,\tag{2a} \label{xy}\
c &= a+b.\tag{2b} \label{ab}
\end{align}
```

加上標籤後, 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)。

如果排版文稿須反覆修改, 則使用 `\label` 與 `\ref` 指令徵引數式較方便, 而且不會出錯。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。假設文稿一開始有 4 條數式是由 `TeX` 自動編號, 接著的兩條式子則以 `\tag` 指令自行編號為 (9.5a) 與 (9.5b), 再下一條數式應為 (9.6)。此時, 我們必須在數式之前調整計數器, 才能得到正確結果:

```
\setcounter{equation}{5}
```

下一條數式出現時, 計數器將加 1, 數式編號即變為 6。

9.7 細節調整

在數式環境下, `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, 符號之間距作適當調整, 上下標選用較小字體, 數學式自動編上號碼等等。但是, 還是有一些情況, `TeX` 的處理結果不見得理想, 必須進一步人為調整。以下分別說明之。

9.7.1 調整符號間距與大小

在數式環境中,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,自動調整之結果可能不理想。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:

$\,$, 加入小空白 (約 1.5pt) $\:$ 加入中等空白 (約 3pt)
 $\!$ 減去小空白 (約 1.5pt) $\;$ 加入大空白 (約 5pt)

指令 $\!$ 可視為是 $\,$ 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。

$\sqrt{2}x$	$\text{\$}\sqrt{2}\,\, x\text{\$}$	$\sqrt{2}x$
$n/\log n$	$\text{\$}n/\! \! \log n\text{\$}$	$n/\log n$
$\iint z dx dy$	$\text{\$}\int\int\! \! \! \int z\,dx\,dy\text{\$}$	$\iint z dx dy$
$x^2/3$	$\text{\$}x^2\!/ /3\text{\$}$	$x^2/3$

依據排版規範,數學式之上下標符號須縮小一些;展示數式之分式的分子與分母亦然。 \TeX 會盡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。

$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}}$	$\text{\[} a_0 + \text{\frac{1}{\displaystyle a_1} + \text{\frac{1}{\displaystyle a_2} + \text{\frac{1}{\displaystyle a_3} + \text{\frac{1}{a_4}}}} \text{\]}$
---	--

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:

`\displaystyle` 展示數學符號之標準大小,
`\textstyle` 隨文數學符號之標準大小,
`\scriptstyle` 第一層上下標之標準大小,
`\scriptscriptstyle` 第二層以後上下標之標準大小。

9.7.2 數學式居中與靠左

依內定值,展示數式將居中排版。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沿一點距離,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。

10 表格

若安排得當，一個表格勝過千言萬語。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 \TeX 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 \TeX 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， \TeX 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表格與超大型表格。

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。



表格

根據專家的意見，排版表格應注意下列要點：

- 表格中勿畫垂直線，
- 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` 巨集套件之說明檔，作者是 Simon Fear。你或許覺得這些意見極端，不過，觀察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，右邊表格則拿掉垂直線，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 指令都難以處理時，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` 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。

10.1 tabular 指令環境

排版表格主要是使用 `tabular` 與 `tabbing` 指令環境。本章首先介紹 `tabular` 指令環境以及相關之巨集套件，最後再說明 `tabbing` 指令環境。在介紹指令之前，我們先簡單說明兩者之差別。

`tabbing` 指令環境的主要功能是將文字與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，我們也可以使用 `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。

`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

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

```

\small
\begin{tabular}{lccc}
\hline
& 央行& 物價& 支出 \\[-2pt]
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 10.1: tabular 指令環境

境。新版 \LaTeX 又提供 `array` 巨集套件以加強 `tabular` 指令環境之功能。除了以上的指令之外，許多專家又寫了各種巨集套件，本章也擇要介紹。

圖 10.1 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與數字之排版位置。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)，表示第一欄文字應靠左排列。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 欄資料要靠右編排，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，欄寬可以自行設定，也可以讓 \LaTeX 計算決定。本例中，欄寬是由 \LaTeX 計算決定，計算方法如下： \LaTeX 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， \LaTeX 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。更簡單的作法是利用 \TeX 之指令,在 `tabular` 指令環境之前加上 `\centering` 指令即可。

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` 指令環境緊接著輸入即可。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 10.1,我們發現第一行標題之文字「央行」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「意大利」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,中文字較英數字為高。故若以 \TeX 的指令排版,通常須作些許調整。欲拉大表格內橫行之間距,可以使用 `\extrarowheight` 指令。譬如,若在 `tabular` 指令環境之前加上:

```
\extrarowheight=2pt
```

即可將表格內之行距加大 2pt。

圖 10.1 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:

```
\begin{tabular}{|l|r|r|r|}
```

排版之後,各欄資料之左右將會出現一直線,如圖 10.2 所示。請注意,我們已將 2-4 欄文字改成靠右排版,並利用 `\extrarowheight` 指令將橫行間距加大 2pt。另外,標題文字之後我們輸入兩道 `\hline` 指令,畫出兩條水平線。

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 10.2: 表格內加上直線

10.1.1 控制欄位間距

表格各欄位間距之內設值為 12pt, 但可以更改之。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`。換言之, `\tabcolsep` 指令設定欄位間距之二分之一。以上之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 10.1 第 2-3 欄之間距若想改為 6pt, 指令為:

```
\begin{tabular}{lc@{\hspace{6pt}}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。如果要消除原有之空白, 我們可加入 `@{}` 指令。

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, 括號內之文字或指令即填入表格中對應的欄位間隔處, 而且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, 圖 10.2 之表格並不符合標準。若使用 `@{...}` 指令, 表格可重新排版如圖 10.3。

本例之表格計有 3 欄, 但第 3 欄為了讓小數點上下對齊, 我們把整數與小數分開排版, 兩者之間再加上小數點。因此, 欄位設定之指令變成: `{llr@{.}l}`, 其中的 `@{.}` 是用於排出小數點。小數點左邊為整數之數字, 設定為靠右 `r`; 小數點右邊為小數之數字, 設定為靠左 `r`。三者合併即為一含小數點之數字。

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定之任何空白, 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 10.5.1 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}{llr@{.}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 10.3: 表格數字上下對齊

10.1.2 booktabs 巨集套件

利用 `\hline` 指令所畫之表格橫線, 粗細都相同。如果我們使用 `booktabs` 巨集套件, 則表格上方及底下可畫出較粗的橫線; 中間則是較細之線條。此巨集指令之作者 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]
\bottomrule[wd]
\midrule[wd]
\cmidrule[wd](trim){a-b}

```

欲畫表格上方之橫線應使用 `\toprule` 指令, `[wd]` 選項用於設定線條粗細。若粗細要定為 1pt, 指令為 `\toprule[1pt]`。若不加選項, 程式會自動選用較粗之線條。`\bottomrule` 指令則用於畫表格底下之橫線。由圖 10.4 可看出, `\toprule` 底下之空白會自動加大一些, `\bottomrule` 上方之空白也會自動調整。

表格內若需畫橫線, 應使用 `\midrule` 指令。但如果表格內之橫線並未貫穿全部欄位, 我們可使用 `\cmidrule(1){2-3}` 指令畫出, 大括號內之數字代表橫線所貫穿之欄位。本例之 `{2-3}` 指示橫線橫跨第 2-3 欄位。圓括號 (1) 選項表示橫線左端要截掉 (trim) 一點。若短線兩端都要截短, 指令為 `\cmidrule(lr){2-3}`。`\cmidrule` 是 `booktabs` 巨集套件之指令。 `tabular` 指令環境也有一個類似的指令, 稱為 `\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。

現假設有一個較大的表格, 共有 5 欄。我們想要在 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}

```

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\\[-2pt]
國家 & 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 10.4: booktabs 巨集套件

我們把左邊橫線的右端截掉一點；右邊橫線的左端也截掉一點。如此一來，兩條短線才能清楚分辨。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。反之，如果在同一欄位上要排出緊接的兩條短線，指令如下：

```
\cmidrule(1){2-3} \cmidrule(1){2-3}
```

圖 10.4 第 2–3 欄上方的一小標題「經濟表現」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，指令為 `\multicolumn{2}{c}{經濟表現}`。

圖 10.4 之表格例子中，欄位指令選項之兩端各加上 `@{}` 指令，其作用除掉表格兩端之空白。這是專業排版之建議。若無此設定，表格兩端會有一點空白。

10.1.3 表格內的文字段落

以上所舉的例子，各欄都是單行的簡短文字或數字，欄寬是由 \TeX 所決定。有時候，表格某一欄的內容是數行的文字，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。

圖 10.5 的表格例子共有 3 欄，前兩欄由 \TeX 依各欄內之數字或文字決定其寬度，第 3 欄以 `p{2.5cm}` 指令設定寬度為 2.5 公分。在固定寬度的表格欄裡，我

項目	分數	評述意見
方法	85	本研究的實驗方法是作者發展出來的。
貢獻	88	從實際應用來看, 本研究很有貢獻。
文字	85	甚佳。

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 & 本研究的實驗方法%
是作者發展出來的。\\[2pt]
貢獻 & 88 & 從實際應用來看,
本研究很有貢獻。\\[2pt]
文字 & 85 & 甚佳。\\
\bottomrule
\end{tabular}\par\smallskip
\parbox{5cm}{以上為虛構,
如有雷同, 純屬巧合。}

```

圖 10.5: 表格內之文字段落

們可以輸入多個段落。本例中第一欄位內僅含一個段落。如果有兩個段落, 第一段結束處應加上 `\par` 指令。請注意, 不能使用換行指令 `\\`, 因為在 `tabular` 指令環境內, 換行指令代表橫行結束。另外, 分段之後, 下一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*{5mm}` 指令, 以內縮 0.5 公分。

圖 10.5 同時說明如何以段落方塊指令 `\parbox` 在表格底下輸入註解。L^AT_EX 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, 註解將會排在表格的右方而不是底下。本例中, `\par` 指令之後先以 `\smallskip` 指令將空白加大一些; 其後再以 `\parbox` 指令編排註解。以上編排表格註解方法的麻煩是, 第一次排版時, 表格有多寬並不清楚, 因此必須來回兩三次才能設定正確的寬度。下一小節所介紹的 `threeparttable` 巨集套件可自動算出表格寬度以排版註解。

圖 10.5 表格例子所使用的 `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 2 欄第一行之數字 “85” 即和第 3 欄文字段落的首行對齊。如果引用 `array` 巨集套件, 我們還可以使用 `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。圖 10.6 是調整橫行位置的一個例子, 其段落是使用 `m{4cm}` 指令排版:

```
\begin{tabular}{lcm{4cm}}
```

我們另以 `\midrule` 指令加上橫線以區隔三列資料。

項目	分數	評述意見
方法	85	本研究的實驗方法是作者發展出來的。
貢獻	88	從實際應用來看, 本研究很有貢獻。
文字	85	甚佳。

圖 10.6: 表格內橫行位置之調整

10.1.4 表格註解

圖 10.5 (頁 168) 說明如何以 `\parbox` 指令排版表格之註解。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 的 `threeparttable` 巨集套件。此巨集套件會自動算出表格寬度, 減少來回調整的麻煩。

巨集套件 `threeparttable` 提供 `threeparttable` 指令環境, 在指令環境內我們仍使用 `tabular` 排版表格, 其上可使用 `\caption` 指令加入表格標題, 其下則以 `tablenotes` 指令環境排版註解, 圖 10.7 是一個例子。排版表格註解時, 表格內先以 `\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, 我們須自行輸入編號或符號。本例含兩個註解, 第一個註解說明一般事項, 第二個註解則是補充說明表格內之「方法」一項。我們在「方法」兩字之後加入 `\tnote{a}`, `tablenotes` 指令環境內則以 `\item[a]` 排出對應的說明。

表格註解之格式可自行設定, 本例在全文設定區作如下之設定:

```
\renewcommand{\TPTtagStyle}{\normalsize\textit}
\renewcommand{\tablename}{表}
```

第 1 行指令指示註解標籤 *a* 是以正常大小之斜體字排版, 第 2 行指令則是在表編號之前加上中文「表」。若無第 2 行指令, 標題將變成 “Table 1:”。表格標題之排版方法, 詳見 10.3 (頁 175) 之說明。

10.1.5 array 巨集套件

以上說明排版表格常用之指令。為強化表格排版之功能, \TeX 又提供 `array` 巨集套件以補充原有之表格指令。在 `tabular` 指令環境內, 表格的畫線指令包括:

表 1: 評估意見總結

項目	分數	評述意見
方法 ^a	85	本研究的方法是作者發展出來的。
貢獻	88	從實際應用來看，本研究很有貢獻。
文字	85	甚佳。

* 詳細意見請見正文。

^a 本方法可推廣到其他領域，值得作進一步研究。

```

\usepackage{threeparttable}
\usepackage{booktabs}
\renewcommand{\TPTagStyle}%
  {\normalsize\textit}
\renewcommand{\tablename}{表}
...
\begin{threeparttable}
\caption{評估意見總結}
\begin{tabular}{lcp{2.5cm}}
\toprule
項目 & 分數 & 評述意見 \\
\midrule
方法\tnote{a} & 85 &
本研究的方法是作者發展出來的。\\[2pt]
貢獻 & 88 & 從實際應用來看，
本研究很有貢獻。 \\[2pt]
文字 & 85 & 甚佳。\\
\bottomrule
\end{tabular}
\begin{tablenotes}\small
\item[*] 詳細意見請見正文。
\item[a] 本方法可推廣到其他領域，
值得作進一步研究。
\end{tablenotes}
\end{threeparttable}

```

圖 10.7: threeparttable 巨集套件

`\hline`, `\vline` 與 `\cline`, 其中, `\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。 `\vline` 指令用於在特定欄位處排版短垂直線, 高度與欄高相同。

另外一個常用的指令是 `\multicolumn`, 圖 10.4 (頁 167) 的例子用它來排版橫跨兩欄位以上之標題文字。表 10.1 列出 `tabular` 指令環境內排版表格之指令, 其中, 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。以下以幾個例子進一步說明表格指令之用法。

首先, 我們介紹 `!{\dots}`, `>{\dots}` 與 `<{\dots}` 指令之用法。以上三個指令是 `array` 巨集套件所提供, 故須於全文設定區引入。 `!{\dots}` 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 10.8 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, `>{\dots}` 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; `<{\dots}` 指令則是將大括號內之指令或文字附加於欄位內容之後。

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,
@{...}	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,
	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,
>{...} [*]	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。

^{*} 須引用 array 巨集套件。

<input type="checkbox"/> Brazil	33.92%
<input type="checkbox"/> China	8.82%
<input type="checkbox"/> India	8.37%

```

\usepackage{booktabs}
\usepackage{array,latexsym}
...
\begin{tabular}{!{\$}\Box$}%
>{\sffamily}lr<{\%}}
\toprule
Brazil & 33.92 \\
China & 8.82 \\
India & 8.37 \\
\bottomrule
\end{tabular}

```

圖 10.8: 表格指令之應用

本例中，附加於前之指令為 `>{\sffamily}`，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 `|` 指令可畫出垂直線，但其粗細無法調整。利用 `array` 巨集套件所提供之 `!{...}` 指令，我們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 10.9 例子中使用兩個 `\vline` 指令，畫出貫穿兩橫列之垂直線。

表 10.1 所列的各項指令，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`，欄位之間距要加大 `2pt`，指令如下：

```

\extrarowheight=2pt
\addtolength{\tabcolsep}{1pt}

```

如果要將表格內之行距加大為原來的 1.2 倍，請在全文設定區加入下列指令：

```
\renewcommand{\arraystretch}{1.2}
```

或者，我們也可以使用 `\extrarowheight` 指令達到類似的效果。

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	

```

\usepackage{array}
\extrarowheight=2pt
...
\begin{tabular}{|l|l|l|}
\hline
單位 & 國立台灣大學 \\
\hline
主持人 & 陳大川~~\vline%
~~322-4433\\
聯絡人 & 張小河~~\vline%
~~733-3344\\
\hline
地址 & 台北市羅斯福路 \\
\hline
\end{tabular}

```

圖 10.9: 表格中短直線

10.2 tabularx 巨集套件

除了 tabular 指令環境之外， \LaTeX 還提供 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*}{.8\textwidth}{lrr}
```

則表格寬度將為行長的 80%，每一欄之寬度仍由 \LaTeX 依各欄之文字或數字自動計算。因為表格寬度已設定為行長的 80%， \LaTeX 會自動把欄距拉大，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，還可以使用 tabularx 巨集套件 (作者 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，但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 10.10 是使用 tabularx 排版表格之例子。tabularx 指令環境須加入兩個設定項，第一項設定為表格寬度，本例選用 \textwidth，也就文字版面之寬度。第二項設定則控制各欄位之排版，我們仍可使用表 10.1 之欄位控制指令，如 l 或 r 等，但選項中至少要有一個 X。本例中，表格全寬為 \textwidth，第 2–4 欄以 r 選項設定靠右編排，其寬度由 \LaTeX 計算。第 1 欄之設定項為 X，欄寬是由表格寬度減去 2–4 欄之寬度以及各欄之間距。為了平衡各欄之間距，本例加入

	平均每人所得	平均每人國富	平均每戶國富
1998	37.9 萬元	460 萬元	1,583 萬元
1999	39.0 萬元	473 萬元	1,602 萬元
2000	40.3 萬元	492 萬元	1,640 萬元

國富毛額之計算包含實物資產與國外資產淨額兩部分，實物資產包括：已登記土地、房屋建築、機械設備、存貨等。

```

\usepackage{tabularx,booktabs}
\newcommand{\tabnote}[1]{\par\medskip\parbox{\textwidth}{#1}}
...
\begin{tabularx}{\textwidth}{@{}Xr@{\hspace{.1\textwidth}}r}
@{\hspace{.1\textwidth}}r@{}
\toprule
& 平均每人所得 & 平均每人國富 & 平均每戶國富\\
\midrule
1998 & 37.9 萬元 & 460 萬元 & 1,583 萬元\\
1999 & 39.0 萬元 & 473 萬元 & 1,602 萬元\\
2000 & 40.3 萬元 & 492 萬元 & 1,640 萬元\\
\bottomrule
\end{tabularx}
\tabnote{國富毛額之計算包含實物資產與國外資產淨額兩部分，
實物資產包括：已登記土地、房屋建築、機械設備、存貨等。}

```

圖 10.10: tabularx 巨集套件

`@{\hspace{.1\textwidth}}` 指令，將間距設定為行長的 10%。若不加入間距指令，1-2 欄之間距會太大，版面效果並不好。

因為表格寬度自行選定，故排版表格註解就很容易。本例中，我們先在全文設定區定義一巨集指令 `\tabnote`，其中使用了 `\parbox` 指令，寬度與表格相同。排版之後，表格註解即與表格本身占有同樣寬度。

當表格之欄位設定以 `X` 指令排版時，欄位內之文字將以 `p{...}` 指令編排（見表 10.1），也就是以 `\parbox[t]{...}` 指令編排。如果希望改以 `m{...}` 指令格式編排，必須在 `tabularx` 指令環境之前重新定義：

```
\renewcommand{\tabularxcolumn}[1]{>m{#1}}
```

經過重新定義，欄位內之文字段落中間會對齊其他欄之文字或數字。

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 \LaTeX 會將6公分平均分給3個欄位。因為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。

10.3 圖表位置與標題

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。 \LaTeX 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。這兩個指令環境合稱為 `float` (浮動版面)；排版時，指令環境內之圖表將置於版面之上下方，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` 指令環境內之標題則加上 `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。

10.3.1 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。`table` 指令環境內應加入排版表格之指令。若是排版圖形，`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。 \LaTeX 選擇圖表的位置時，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。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，如此之安排也不適當。 \LaTeX 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 [Mittelbach and Goossens \(2004\)](#)，第6章。在計算圖表之位

置時, \LaTeX 會參考一些控制值。有專家認為依 \LaTeX 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}
```

設定值須定義於全文設定區內。

經以上之設定, 若對於 \LaTeX 的之選擇仍不滿意, 可進一步在 `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。

如果你設計的版面特別, 可在下位置選項變數時加入 `!`, 要求 \LaTeX 儘可能遵照你所下的指令選項置放圖表。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。

10.3.2 圖表標題

浮動版面指令環境之內可以使用 `\caption` 指令排版標題。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，上方空白反而不是必要的。若以上內設之間距不適用，可以重新設定。譬如，若標題是排版於圖表下方，在全文設定區加入下列指令即可重新設定間距為 8pt:

```
\abovecaptionskip=8pt
\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}{8pt}%
\caption}
```

定義指令後，圖標題仍以 `\caption` 指令編排，表標題則改用 `\topcaption` 排版。

如上所述，`table` 與 `figure` 指令環境唯一的差別是在於標題文字不同。中文稿中，表格標題應以「表 3:」起頭；圖標題則以「圖 9:」起頭。標題文字欲改為中文，須在全文設定區加入以下兩道指令:

```
\renewcommand{\tablename}{表}
\renewcommand{\figurename}{圖}
```

`\tablename` 原始定義為 Table，以上指令將之改為「表」。如果要選用圓體字，則指令最後應改為 `{\ctxfr 表}`。

10.3.3 caption 巨集套件

使用 `\caption` 指令排版標題時，圖表會自動編上號碼。在 book 文稿類別下，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 `caption` 巨集套件第 3 版，作者為 Axel Sommerfeldt。請注意，`caption` 巨集套件第 3 版與上一版之指令並不相容。

簡單的設定方法是引入 `\usepackage{caption}` 指令之後，以 `\captionsetup` 指令設定格式。圖表之標題將以內定之設計排版。如果標題要以中文圓體及英文 sans serif 字體排版，全文設定區應作如下之設定:

```

\usepackage{caption}
\captionsetup{font=sf,font=small}
\renewcommand{\figurename}{\ctxfr 圖}
\renewcommand{\tablename}{\ctxfr 表}
\ctxfdef{\caption}{\ctxfr}

```

第2行指令選用小一級之 `\sffamily` 字體。不過, \TeX 之字體指令只能改變英文與數字, 中文須另以指令變更。第3–4行設定將「圖」「表」二字以圓體字排版。第5行以 `\ctxfdef` 指令設定全文 `\caption` 指令內之中文全部以圓體字編排。

如果你進一步要把標題加上藍色, 可在全文設定區作如下之設定:

```

\usepackage{caption,color}
\DeclareCaptionFormat{captionblue}{\color{blue}#1#2#3\par}
\captionsetup{font=sf,format=captionblue}

```

第1行引入 `caption` 與 `color` 巨集套件, 第2行指令設定 `captionblue` 格式, 將標題文全部以藍色排版。定義中之 `#1#2#3` 分別是標題標識(含編號), 區隔符號(冒號), 以及標題文字, `\color{blue}` 指令將整個標題段落變成藍色。第3行指令引用 `captionblue` 格式。`\color{blue}` 指令之詳細說明, 請見第12章。

表10.2列出 `caption` 巨集套件較常用之選項。當表格標題超過一行時, 你可加入 `format=hang` 之設定, 讓標題以 `hang-indent` 方式排版, 亦即第2行文內字會內縮。依內定值, 標題標識與編號之間是以冒號區隔, 但我們可改為句點: `labelsep=perild`。 `justification=` 選項則設定標題要居中, 靠左, 或靠右。不管標題是選排靠左或靠右, 如果標題文字只占一行, 我們可設定成居中排版, 方法是加入 `singlecheck=on` 選項。若同時加入以上之設定, 指令為:

```

\captionsetup{format=hang,labelsep=period,singlecheck=on,font=sf}

```

其中, `singlecheck=on` 為內定值, 事實上可以省略。各設定指令詳細說明, 請見 `caption` 之說明檔。

10.4 引述表格

\TeX 提供 `\label`, `\ref`, 與 `\pageref` 指令以引述數學式, 章節, 以及圖表。本書排版表10.2時, 標題之排版指令如下:

```

\caption{\textsf{caption} 巨集套件常用選項}\label{tabcap}

```

表 10.2: caption 巨集套件選項

format=	default, hang
labelsep=	colon, period, space
justification=	justified, centering, centerlast, centerfirst, raggedright,raggedleft
singlecheck=	true, yes, on
font=	small, large, sf, it, bf, ...
labelfont=	small, large, sf, it, bf, ...
textfont=	small, large, sf, it, bf, ...
margin=	長度

`\caption` 指令之後加上 `\label{tabcap}` 定義標籤為 `tabcap`。定義了標籤之後，文稿內任何地方可以使用 `\ref{tabcap}` 指令引述此表；或以 `\pageref{tabcap}` 引用頁碼。

舉例言之，行文中欲排版下列句子：

如表 10.2 (頁 179) 所示 ...

輸入指令如下：

如表`\Z\ref{tabcap}` (頁`\Z\pageref{tabcap}`) 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。以上指令中特別加上 `\Z` 以控制中文與數字之間距。若不加上 `\Z`，仍可得到結果，但間距會嫌太小。

10.5 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，有一些表格排版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。表 10.3 (頁 166) 曾說明如何以 `@{.}` 指令排版上下對齊之小數點，但指令稍嫌麻煩。欲讓小數點上下對齊，可使用 `dcolumn` 巨集套件，作者是 David Carlisle。

10.5.1 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。利用 `dcolumn` 巨集套件，我們可直接使用其內建之 `D` 指令設定小數點上下對齊；或者，我們也可以自行定義指令。內建指令 `D` 之格式如下：

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 10.11: dcolumn 巨集套件

D{鍵入符號}{輸出符號}{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 `\newcolumntype` 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:

```
\newcolumntype{.}[1]{D{.}{.}{#1}}
```

以上指令定義之指令取名為英文句點,而且鍵入與輸出符號也都是英文句點。使用指令時,指令選項中須填入小數點位數。

圖 10.11 的例子中,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,應填入最大的位數。圖 10.11 的表格是以底下之指令排版:

```
\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

年期	產量
1991–1992	934,345
1992–1993	1,134,045

```

\usepackage{dcolumn,booktabs}
\newcolumntype{a}[1]{D{-}{-}{#1}}
\newcolumntype{b}[1]{D{.}{,}{#1}}
...
\begin{tabular}{a{4}b{3}}
\toprule
\multicolumn{1}{c}{年期} &
\multicolumn{1}{c}{產量} \\
\midrule
1991-1992 & 934.345 \\
1992-1993 & \mbox{1,134}.045 \\
\bottomrule
\end{tabular}

```

圖 10.12: 表格數字上下對齊

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。

表格第 3 欄上下共有 3 個數字，第 2 個數字 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；欄位之寬度除了容納整數之外，還可容納 3 位小數。此外，第一個數字之後有百分比符號。因為百分比符號並非數字，如果直接鍵入，排版時小數點無法對齊。本例中，我們使用 \TeX 的指令 `\rlap` 來排版百分比符號。此一指令會在 46.7 的右邊排出百分比符號，但是 \LaTeX 在作上下對齊之排版動作時，不會把百分比符號納入考慮。與 `\rlap` 對應的指令是 `\llap`；前者處理字詞右邊之文字符號，後者則處理左邊的文字符號。

此欄如果改以 `.{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 \LaTeX 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。

有人可能覺得奇怪，為何要區別鍵入與輸出符號？圖 10.12 的例子說明此種區別之用途。此表格中有兩欄數字，其中，右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。

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 10.13: multirow 巨集套件

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。排版時，小數點會改為逗號，並且上下對齊。事實上，第二個數字是輸入為 `\mbox{1,134}.045`，原因是表格內若有逗點， \TeX 會自動在其後加入一空格，排版成 “1, 134.045”。故我們使用 `\mbox` 指令以避免此問題。

10.5.2 橫列文字對齊

圖 10.13 的表格中，左欄之「投資」對應右欄之「政府部門」與「民間企業」。排版時，「投資」若對準右欄上下兩列資料中間，版面效果較佳。欲排版此種表格，可使用 `multirow` 巨集套件，作者為 Terry Leichter 與 Piet van Oostrum。

`multirow` 巨集套件，提供 `\multirow` 指令，格式如下：

```
\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，詳見巨集套件之說明。

10.5.3 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。此一指令取自 [Goossens et al. \(1997, 頁 49\)](#), 詳細說明請見原書。簡單言之, 我們利用 `graphicx` 巨集套件之 `\rotatebox` 指令將可放大縮小之數學符號作某一角度之旋轉。因此, `\bpara` 之符號可放大, 也可旋轉特定角度。巨集指令 `\bpara` 必須定義於全文設定區, 指令內容如下:

```
\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 10.14 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 16pt。第 2 個大括號長度為 14pt, 但該欄位內並無任何文字, 經過幾次嘗試, 我們發現若右移 33pt, 大括號可以上下對齊, 故指令為 `\bpara{33}{0}{0}{14}`。其餘兩個大括號之指令也是反覆調整後決定的。

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 9.14 (頁 150) 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}
```

圖 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	桃園廳	
新竹廳		

```

\usepackage{graphicx,booktabs}
\renewcommand{\tablename}{圖}
...
\begin{table} \centering
\caption{行政區域劃分變遷}
\begin{tabular}{l@{\hspace{4pt}}%
l@{\hspace{4pt}}l}
\toprule
~1905& ~1915& ~1920\\
\midrule
台北廳 & & \\
基隆廳 \bpara{0}{2}{0}{16} & & \\
& 台北廳 & \\
深坑廳 & & \\
\bpara{33}{0}{0}{14} & 宜蘭廳 & \\
& \bpara{0}{0}{0}{35} & 台北州\\
宜蘭廳 & & \\
桃園廳 \bpara{0}{0}{0}{16} & & \\
& 桃園廳 & \\
新竹廳 & & \\
\bottomrule\end{tabular}
\end{table}

```

圖 10.14: 表格內之大括號

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, 本例定為 0.4pt。利用以上定義, 即可在表格中加入斜線。圖 10.15 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清楚的設計。

10.6 彩色表格

排版的目的是精簡地傳達資訊, 因此表格上應避免花花綠綠的色彩。不過, 表格內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

$x \backslash y$	y_1	y_2
1	34	55
2	25	45

```

\usepackage{graphicx}
...
\begin{tabular}{l@{r}rr}
&  $y_1$  & & \\
& &  $y_1$  &  $y_2$  \\
\hline
1 & 34 & 55 \\
2 & 25 & 45
\end{tabular}

```

圖 10.15: 表格內斜線

表格中加入灰階與色彩, 可使用 David Carlisle 之 `colortbl` 巨集套件。

圖 10.16 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]
```

欲將某一橫列加上灰階, `\rowcolor` 指令須加在該列最前方。指令內之 *color model* 是 `color` 巨集套件選定色彩的方式, 詳見第 12 章之說明。

簡單來說, 排版時我們可使用灰階色或彩色。若以灰階色排版, *color model* 為 `gray`, *color* 則代表灰階之深淺度, 可選用 0 到 1 之間的數字。如果要選用黃色, *color model* 選項可略過, *color* 選項中則填入 `yellow`。本例中, 我們使用灰階 `gray`, 灰階之深淺度為 0.85。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{pureblack}` 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。

通訊錄		
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}{\ctxfbb 通訊錄}
\par\medskip
\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 10.16: 灰階表格

對應於 `\rowcolor` 指令, `\columncolor` 指令可將表格特定欄加上色彩或灰階, 指令如下:

```
\columncolor[color model]{color}[left overhang][right overhang]
```

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 (表 10.1, 頁 171) 之功能。圖 10.17 將表格第 1 欄加上黃色背景, 其方法是在 `tabular` 指令環境中使用下列指令:

```
\begin{tabular}>{\columncolor{yellow}}cccc
```

上述指令中, `yellow` 是 \TeX 內定之顏色, 不過, 本書是以黑白列印, 因此黃色之背景是以灰階出現。

圖 10.17 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 10.17: 彩色表格

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 10.18 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。

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。表格標題欲以彩色或灰階排出,可使用 `\textcolor` 指令,這也是 `color` 巨集套件之指令;請見第 12 章之說明。

10.7 超大型表格

超大型表格是指表格寬度大於文字版面寬度,或者表格高度超過文字版面高度。如果超出之尺寸不大,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。以下分別介紹表格旋轉與排版超長表格的方法。

10.7.1 旋轉表格

欲旋轉表格至少有三種方法,第一種方法是使用 `\rotatebox` 指令,這是 `graphicx` 巨集套件所提供之指令。第二種方法是使用 `rotating` 巨集套件。第三種則是

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}
\definecolor{light}{gray}{.85}
\fbboxsep=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 10.18: 圖表加上灰階背景

使用 `lscap` 巨集套件。`\rotatebox` 指令較適合旋轉小表格或文字，而且可旋轉任何角度，指令之細節請見 12.1 節 (頁 213) 之說明。`rotating` 巨集套件可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或圖形。`lscap` 巨集套件則可進一步旋轉跨頁之超長表格，其指令將於下一小節介紹。

`rotating` 巨集套件提供 `sidewaysfigure` 與 `sidewaystable` 兩個指令環境；前者用於旋轉圖形，後者用於旋轉表格，作者是 Sebastian Raetz 與 Leonor Barroca。指令之使用方法很簡單，只要把圖表之排版指令輸入於指令環境內即可。圖 10.19 是 `sidewaysfigure` 指令環境的一個例子。若是以 book 文稿類別排版，因為單雙頁之版面不同，圖表會轉成適合由版面外沿閱讀之方向。

10.7.2 超長表格

若表格高度超過文字版面高度，可使用 `longtable` 巨集套件排版，作者為 David Carlisle。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` 指令環境替換為 `longtable` 指令環境，原來之表格即自動拆為兩部分以上，分別排版於兩頁或多頁中。須注意的是，若使用 `longtable` 指令環境，排版須執行 `latex` 兩三次，才能得到正確結果。

圖 10.20 為排版超長表格的一個例子。`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」。排版表格時，如果使用 `table` 指令環境，我們即可以 `\caption` 排版

表 1: 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。

```

\usepackage{rotating, threeparttable, booktabs}
\newcommand{\tablename}{表}
...
\begin{sidewaysfigure}
\begin{threeparttable}
\centering
\caption{淡水與基隆港出口貨物量}
\begin{tabular}{lrrrrrrrr}
\toprule
& 1899 & & 1900 & & 1901 & & 1902 & & 1903 & & 1904 & & 1905 & & 1906 & & 1907 & & 1908 \\
\midrule
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 \\
\bottomrule
\end{tabular}
\end{threeparttable}
\begin{tablenotes}
\item[*]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。
\end{tablenotes}
\end{sidewaysfigure}

```

圖 10.19: rotating 巨集套件

```

\usepackage{longtable,booktabs}
...
\begin{longtable}{@{}lrrrrr@{}}
\caption{台灣長期物價指數}\label{longtable}\\
\toprule
& & 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 \\
\midrule
\endfirsthead
\multicolumn{6}{l}{\ctxfk 承接上頁}\\[2pt]
\toprule
& & 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 \\
\midrule
\endhead
\midrule
\multicolumn{6}{r}{\ctxfk 續接下頁}
\endfoot
\endlastfoot
1996 & --~ & --~ & --~ & 60.31& 59.16\\
1997 & --~ & --~ & --~ & 66.88& 55.66\\
...
1937 & 135.98 & 187 & 131.49& 97.84& 432.91\\
1938 & 138.26 & 216 & 165.05& 125.06& 468.77\\
\bottomrule
\end{longtable}

```

圖 10.20: longtable 巨集套件

表格標題。longtable 指令環境之性質與 table 不同，但我們仍可使用 \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。但請特別注意，\caption 指令之後須加上換行指令（圖 10.20 第 4 行），否則排版時會出現錯誤訊息。我們也可以引用 caption 巨集套件設計標題。

文稿內開始輸入 longtable 指令環境時，表格即將從該處開始排版。而指令環境結束之處，其下即接著排版後續之文字。如果希望長表格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由 \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。

由本例子可知，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，因為表格超過一頁，每一頁都須排版欄位標題，但是，表格第一頁之欄位標題可能與其下各頁之標題不同。譬如，表格第 2 頁開始，上方可能須排版「承接上頁」四個字。同理，第 1 頁表格尾端之排版方式可能也與其下

各頁不同。譬如，若表格有3頁，第1-2頁底端可能須排版「續接下頁」四個字，但最後一頁則無此必要。

為處理以上問題，`longtable` 指令環境提供 `\endfirsthead` 與 `\endhead` 等指令以排版欄位標題。圖 10.20 的例子中，`\endfirsthead` 以上之指令與文字是用於排版表格第一頁上方之欄位標題及橫線；而 `\endhead` 以上 (`\endfirsthead` 以下) 之指令與文字，則是排版表格第2頁開始之欄位標題文字。同理，`\endfoot` 以上 (`\endhead` 以下) 的指令是排版表格每一頁 (不含末頁) 下方之版面。本例以 `\midrule` 畫一細橫線；並加上「續接下頁」四個字。最後，`\endlastfoot` 以上之指令則用於排版表格末頁底下之文字或符號。本例為空白，因此表格末頁底下即無任何添加資訊。表格最末端之橫線是由表格排版指令倒數第2行之 `\bottomrule` 所排版。

表 10.3: 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
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

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
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

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，而不是表格的末端。如果要在表格最末端加上註解，可利用 `\endlastfoot` 指令。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。

如果表格又長又寬，我們還可以使用 `lscape` 巨集套件將 `longtable` 所排版之跨頁表格旋轉 90 度，作者也是 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 10.3 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}
```

上面所介紹的 `rotating` 巨集套件也可以旋轉文字圖表，但以一頁為限。`lscape` 巨集套件則可連續旋轉數頁。因此，如果你要旋轉 `pkglongtable` 之長表格，應使用 `lscape` 巨集套件。

	大	中	小	
牛肉麵	120元	100元	80元	<code>\begin{tabbing}</code>
搾醬麵	60元	50元	40元	<code>\ctxfk</code>
酸辣湯	40元	30元	20元	<code>\hspace*{1.3cm}\=~~大\qqad\= %</code>
				<code>~~中\qqad\=~~小\\</code>
				<code>\ctxfm</code>
				<code>牛肉麵 \> 120元 \> 100元 \>80元\\</code>
				<code>搾醬麵 \>~~60元 \>~~50元 \>40元\\</code>
				<code>酸辣湯 \>~~40元 \>~~30元 \>20元</code>
				<code>\end{tabbing}</code>

圖 10.21: tabbing 指令環境

10.8 tabbing 指令環境

除了 `tabular` 指令環境排版之外， \TeX 尚提供 `tabbing` 指令環境以供排版表格。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，很適合用 `tabbing` 指令環境來排版。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 10.21 之表格例子共有 4 欄，各欄之間距須自行設定。本例第 3–4 行指令除了排版標題之外，同時使用 `\=` 指令設定欄寬；行末之 `\\` 表示此行指令結束。欄寬設定之後，其下即可以 `\>` 指令列表。譬如，第 1 欄之數字排版之後，可使用 `\>` 指令跳至第 2 個欄位。

欄寬是決定於 [Tab] 之位置，故設定欄寬也等於是設定 [Tab] 之位置。本例之欄寬是以下列方式設定。首先，第 1 欄之標題空白，但估計排版該欄位之文字需 1.3 公分，故以 `\hspace*{1.3cm}` 直接設定欄寬。其後之 `\=` 指令即標示第 1 個 [Tab] 位置。接著排版第 1 欄之欄位標題「大」字，其後以 `\qqad` 指令拉開一點空白；再加上 `\=` 指令，這就是第 2 個 [Tab] 之位置。因為第 2 欄位底下之數字較標題為寬，故我們在「大」字之前加上 3 個空格指令 `~~~`，以免第 1 欄之數字太靠左邊。第 3–4 欄之寬度也是以類似的指令設定。

經過以上之設定，第 6 行開始輸入表格第 2 行之文字與數字。首先，我們鍵入第 1 欄之「牛肉麵」三個字，之後使用 `\>` 指令跳至第一個 [Tab] 位置，再鍵入「120元」，餘此類推。在 `tabbing` 指令環境中，文字與數字是緊貼著 [Tab] 位置排出。本例第 3 行之第 2 欄數字為「60元」，其上方為「120元」。為了使個位數上下對齊，輸入「60元」時，我們在其前加上兩個調整空白之指令 `~~`。

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 10.22: 以樣本行設定欄寬

本例中，每一行都有第 4 欄文字。若某欄位空白，連續兩個 `\>` 指令即可跳到再下一個 [Tab] 位置。同理，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。

10.8.1 以樣本行設定距離

圖 10.21 的表格例子中，設定欄寬的第一行文字是表格的一部分。圖 10.22 是以一樣本行 (sample line) 設定欄寬之例子，排版時樣本行並不出現，第 2 行指令為樣本行，行末須加上 `\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 欄是以「美國道瓊數」5 個字作為欄寬；第 2 欄則以 9 個“2”字作為欄寬。這些文字與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，排版減號應進入數學模式：`-$-1.1`，或者 `-$-1.1$`。如果未進入數學模式，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；但結果並不十分理想。

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。如果要將文字串之右端擺於 [Tab] 位置，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，指令左邊的文字將排於 [Tab] 位置之左，其右再接著排出文字串右半部分之文字。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 10.23 是一個例子。此例中，我們設定兩個 [Tab]，第 1 個 [Tab] 位置距離版面左沿 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，本例設定為 0.2mm。

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 10.23: tabbing 指令環境

與 \’ 指令相對的是 \‘, 其功能是将排版文字往右推擠到版面右沿。如果某文字串之前加上 \’ 指令, 而且其後沒有 \= 或者 \> 指令, 則此文字串將居右 (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 56)。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。

10.8.2 其他控制指令

圖 10.23 之表格例子中, 第 1 欄文字是從第 1 個 [Tab] 位置開始編排, 故每一行之前端都須加上 \> 指令。為了簡化輸入, 我們可以在第一行前端改輸入 \+ 指令。此一指令之作用等於是在本行與以下各行前端加入 \> 指令, 故第 2 行開始, 各行前端就不須再輸入 \> 指令。

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 \+ 指令從該行即取消。

11 引用外製圖形

圖形是傳達資訊的有效工具，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，L^AT_EX 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 P_IC_TE_X、L^AT_EXCAD 等，直接使用 L^AT_EX 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。本章以介紹 PostScript 圖形為主，但也兼及 PDF 及一些常用的描邊圖形格式，如 PNG 與 JPEG。關於引用 PostScript 圖形的方法，Reckdahl (1997) 有很詳細的討論，值得參考。

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」圖形是一個點一個點地在白紙上畫出圖形來。Windows 內附之小畫家軟體或者 Photoshop 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。幾個有名的繪圖軟體，如 Corel Draw, Visio, Adobe Illustrator 等，所產生的則是描邊圖形。

就使用彈性與排版品質而言，描邊圖形是最好的選擇。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。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。

在網路時代, JPEG 與 PDF 格式之圖形檔日漸普及。譬如,數位相機之檔案通常是 JPEG 格式,而網路文稿常儲存為 PDF 格式。JPEG 是描點圖形,PDF 可能是描邊,也可能描點。引用外製圖形的一個重點是,圖形之規格影響你的排版方式。第 3.3 節(頁 40)說明,我們可以透過 latex + dvips 產生 PS 檔案,也可以經由 pdflatex 產生 PDF 檔案。如果外製圖形是 EPS 檔案,你必須利用 latex + dvips 排版。換言之,若以 pdflatex 排版,而文稿內引用了 EPS 檔案,排版程式將發出錯誤訊息。

圖形規格與排版方法之間的關係可簡單歸納如下:

- EPS 圖形: 僅能使用 latex + dvips 排版,
- JPEG 與 PNG 描點圖形: 可使用 latex + dvips 排版; 也可以使用 pdflatex 排版,
- PDF 圖形: 僅能使用 pdflatex 排版。

另外,我們也可以使用 dvipdfmx 將 latex 之排版結果輸出為 PDF。此一軟體目前正積極發展當中,依我們的測試,它也可以處理 EPS, PDF 與描點圖形。以下,我們依序說明引用 PostScript 圖形,描點圖形,以及 PDF 圖形的方法。

11.2 引用 PostScript 圖形

PostScript 圖形規格是美國 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,印表機有能力將圖形印出來。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;而具有列印 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,但價錢較高。

幸運的是, 透過 Ghostscript 軟體之運作, 一般的 HP 或 EPSON 雷射印表機或噴墨印表機即具備 PostScript 能力。Ghostscript 可免費使用, 它是由 L. Peter Deutsch 等人所發展, 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。

PostScript 與 T_EX 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, 硬碟內即儲存一份 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圖形檔格式, 稱為 Encapsulated PostScript, 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, 主要的差別在於 EPS 檔案內儲存有標識圖形大小的座標, 一般稱之為 bounding box, 底下將進一步說明。

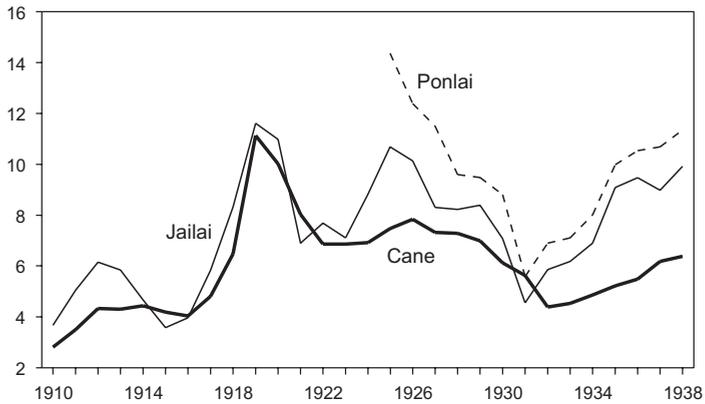
引用 EPS 圖形的最大困難不在於如何下指令控制, 而是在於產生正確的 EPS 檔案。很多畫圖軟體所產生的 EPS 圖形檔都不正確, 引入文稿中即出現問題。本節首先說明引用外製圖形的指令, 下一節再介紹幾種繪製 EPS 圖形的方法。

11.2.1 graphicx 巨集套件

引用外製圖形, 應使用 graphicx 巨集套件之 `\includegraphics` 指令。此一指令可用以引入 EPS 圖形, 也可以引入 PDF 與 JPEG 等格式之圖形。T_EX 處理圖形的方法是透過預覽/列印程式。因此, 使用 graphicx 巨集套件時, 我們須指定使用那一種驅動程式。最常用之驅動程式為 `dvips`, `pdftex`, 以及 `dvipdfm`。如果你使用前兩者, 排版程式會自動判斷, 故引入巨集套件時不須以選項設定。但是, 如果是使用 `dvipdfm` 或 `dvipdfmx` 程式, 引入巨集套件時須加入 `dvipdfm` 選項:

```
\usepackage[dvipdfm]{graphicx}
```

圖 11.1 是引入外製圖形的一個例子, 圖形檔名為 `cx3-rice.eps`。因為是以 latex + dvips 排版, 故指令第 1 行引入 graphicx 巨集套件時未加入驅動程式選項。`\centering` 指令的功能是使圖形居中編排。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。



```

\usepackage{graphicx}
...
\begin{figure}
\centering
\includegraphics[width=0.8\textwidth]{cx3-rice.eps}
\caption{台灣的米價與蔗價}
\end{figure}

```

圖 11.1: 台灣的米價與蔗價

本例引用圖形時，加入 `width=0.8\textwidth` 之設定，將圖形之寬度調整為行長的 80%。請注意，寬度調整時，高度也同比例調整。欲調整圖形之大小，我們還可以使用下列之指令：

- `height` 圖形高度，
- `totalheight` 圖形全高，
- `width` 圖形的寬度，
- `angle` 圖形旋轉 (反時鐘方向) 角度，
- `scale` 圖形放大 (或縮小) 之倍數。

想像把圖形放在一長方形中，左下角稱為基準點 (reference point)。由基準點起，可以算出圖形的高度 (`height`)，寬度 (`width`) 與深度 (`depth`)；其中，深度是指基準點以下之長度。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，基準點仍為原來位置，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。

若要將圖形寬度拉大為與行長相同，指令如下：

```

\includegraphics[width=\textwidth]{cx3-rice.eps}

```

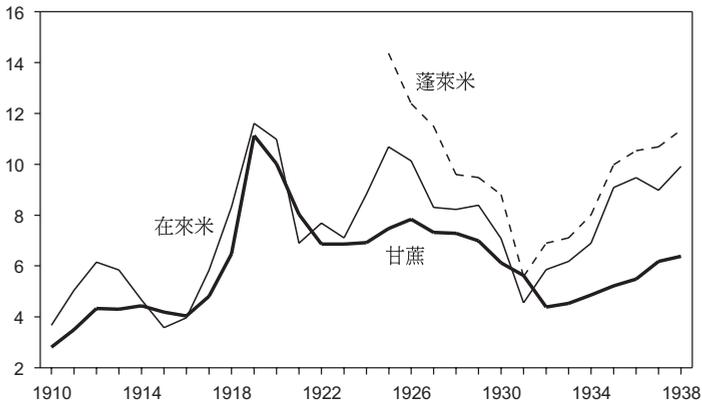


圖 11.2: 台灣的米價與蔗價

如果要將圖形縮小為原尺寸的 80%，設定指令為 `scale=0.8`，圖形的寬度與高度將同比例變動。若要作不同比例之調整，須同時設定高度與寬度，譬如：

```
\includegraphics[width=10cm,height=7cm]{cx3-rice.eps}
```

以上指令設定圖形高度為 7 公分，寬度為 10 公分。不過，高度與寬度若作不同比例之縮放，可能會讓圖形顯得「不自然」。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 圖形內加入中文或數式

圖 11.1 含有三條線，都是以英文標識名稱。若是排版中文稿，以中文標識線條會比較清楚。一般的商業軟體都有輸入中文的能力，因此，圖形要加入中文標識並不困難。我們將原圖形之英文標識改為中文，結果如圖 11.2。此圖是以 Excel 畫出圖形，貼入 Corel Draw (第 9 版) 軟體內，鍵入中文標籤，再輸出 (export) 為 EPS 檔案。輸出時，Export text as 之選項是選擇 Text，並勾選 include fonts。

以一般的商業軟體畫圖時，我們可以直接加入中文。但圖形內之中文若使用 Windows 之細明體或楷體，其字型與正文 `cmTeX` 之字型並不搭配。`cmTeX` 有 5 套 TrueType 字型，可供 Windows 軟體使用。這 5 套字型檔置於光碟 `\util\TrueType` 子目錄內，安裝方法如下：



TrueType

開始 > 控制台 > 字型 > 檔案 > 安裝新字型

安裝之後，Corel Draw 或 Excel 等 Windows 軟體即可使用這些字型。圖 11.2 之線條標籤即使用 `cmTeX` 明體字。

11.2.3 psfrag 巨集套件

雖然多數的商業軟體可以輸入中文, 但仍有許多科學軟體無法處理中文。另外, 如果你要在圖形中加上複雜的數學式子, 不管是商業軟體或科學軟體都很麻煩。解決以上問題的方法是使用 `psfrag` 巨集套件 (由 Michael C. Grant 與 David Carlisle 共同發展)。`psfrag` 巨集套件的原理很簡單。繪製圖形時, 我們在圖形鍵入英文字元; 引入圖形時, 我們再下指令將英文字元轉換為中文字串或數學式。

以圖 11.1 為例, 首先圖形內之英文字須改為單一字母, 例如, Ponlai 改為 P, Jailai 改為 J, Cane 改為 C。要把 J 替代為「在來米」, 指令為:

```
\psfrag{J}{在來米}
```

排版結果如圖 11.3 所示。本例子把 J 替代為「在來米」, C 替代為「甘蔗」, P 替代為一串數學式。若替代之字串可能超過一行, 則應置於 `\parbox` 指令環境內。本例之數學式置於 `\parbox` 指令環境內, 不過其長度並未超過一行。

以 `\psfrag` 指令作字元之替代時, 替代到圖形內之字串的字體與字級可以自行設定。另外, 為了精確置放替代字串, 我們可以加入調整位置之指令。以圖 11.3 為例, 想像原來圖中之字母 J 與替代字串「在來米」三個中文字各有一長方形外框。每一個外框可以用英文字母 t, b, l, r 之組合設定基準點: 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, tl 代表長方形外框左上角, bl 為左下角基準點, b 代表下方中間點。

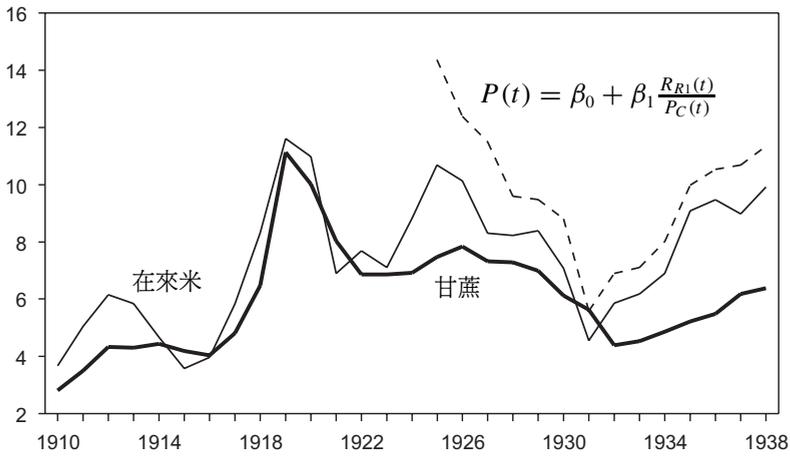
若要把替代字串下方之中心點置於原字元之上方中心點, 指令為:

```
\psfrag{J}[b][t]{在來米}
```

換言之, 第一個方括號標識替代字串外框之基準點, 第二個方括號則標識原始字元之外框基準點。如果不標識位置, 則原始字元與替代字串之基準點將自動設為 [bl], 亦即原始字元與替代字串之左下角將對準。

除了設定位置之外, 替代字串還可以放大、縮小或旋轉。旋轉指令是指替代字串外框之基準點沿逆時針方向旋轉之角度。繼續使用以上的例子, 若字體不放大, 亦即放大倍數等於 1, 而字元要旋轉 90 度, 指令為:

```
\psfrag{J}[b][t][1][90]{在來米}
```



```

\usepackage{graphicx,psfrag}
...
\centering
\psfrag{J}{\small 在來米}
\psfrag{C}{\small 甘蔗}
\psfrag{P}{\parbox{.4\textwidth}{
  $P(t)=\beta_0+\beta_1\frac{R_{R1}(t)}{P_C(t)}$}}
\includegraphics[width=0.9\textwidth]{cx3-ricePS.eps}
\end{figure}

```

圖 11.3: 台灣的米價與蔗價

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* 為原來圖形中之字元，*PSpn* 標識原字元外框之基準點，*pn* 標識替代字元之參考位置，*scale* 為放大倍數，*rot* 表示旋轉之角度，*text* 則代表替代字串。

使用 `psfrag` 巨集套件時有幾個應注意的事項：

- `\psfrag` 指令只能用於 PostScript 描邊圖形。描點圖形 JPEG 或者 PDF 圖形皆不適用，
- 圖形內的英文字元必須是單一字母，若使用兩個字母以上，如 Cane，可能無法替代成功。



psfrag 巨集
套件

- 必須以 latex + dvips 排版, 再以 GSview 預覽或列印, YAP 無法顯示正確結果; pdflatex 也無法處理 \psfrag 指令。
- 下 \psfrag 指令時, 須確定指令中之字元與 EPS 檔案內之字元完全相同, 包括大小寫、空白等。
- 繪圖時, 圖形內的英文與數字應選用英文字型, 如 Times New Roman 或 Arial。若選用新細明體或楷體內之英文字作為字元, \psfrag 指令無法運作。
- 輸出 EPS 檔案時, 英文與數字必須存為字元, 不可存為圖形。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。
- 替代之字串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。

11.3 輸出 EPS 圖形

在 Windows 系統下, 許多軟體都可以繪製圖形。譬如, Excel 試算表可由輸入之資料畫出圖形; Corel Draw 與 Adobe 公司的 Illustrator 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有輸出 EPS 檔案的能力。例如, 以 Corel Draw 畫圖之後, 可選擇將結果輸出為 EPS 檔案; 但是, Microsoft 公司的 Excel 軟體並無此功能, 較新版的繪圖軟體 Visio 也不能直接輸出 EPS。

若軟體本身提供輸出工具, 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, 但輸出工具之功能卻大有問題。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。譬如, Excel 軟體畫出之圖形可以先剪貼到 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輸出為 EPS, 那就是利用 PostScript 印表機驅動程式將圖形「列印」為 EPS 檔案, 下一小節將說明使用的方法。

11.3.1 PostScript 印表機驅動程式

PostScript 圖形語言是由 Adobe 公司所發展, 該公司也免費提供 PostScript 印表機驅動程式。欲使用驅動程式輸出 EPS 圖形檔, 你的電腦中必須有 PS 印表機驅動程式以及 perl 程式。如果原來安裝 c\TeX 時, 你並未選擇安裝這兩個程式, 請依以下方法安裝

- 安裝 PostScript 印表機驅動程式

在 cwTeX 光碟中, $\backslash\text{util}$ 子目錄下有 winstcht.exe , 直接執行即開始安裝。安裝過程中, 電腦會訊問:「你是否要列印測試頁」, 應選「否」; 其他依內定值即可。安裝完成之後, 由

開始 > 印表機和傳真

除了原已安裝之印表機之外, 會出現新的一項 Generic PostScript Printer, 列印 PostScript 檔案時應選用此印表機。

- 安裝 perl 程式

光碟 $\backslash\text{util}$ 子目錄下有 $\text{ActivePerl-561.exe}$, 直接執行即開始安裝。

以下以 Excel 軟體為例, 說明如何產生 EPS 檔案。啟動 Excel 軟體, 並開啓一檔案, 選擇所要列印的範圍, 按「列印」, 印表機須選用 Generic PostScript Printer, 並勾選「列印至檔案」。按「確定」之後, 輸入檔名, 硬碟內即產生一 PS 檔案。依內定值, 此檔案會以 .prn 為附加檔名, 如 PStest.prn , 請將之改名為 PStest.ps 。要把此一圖形引入 L\TeX 檔案中, 我們須把 PS 檔案轉換為 EPS (Encapsulated PostScript)。EPS 檔案與 PS 檔案不同的地方在於前者記錄有標識圖形大小的座標。而驅動程式所輸出的檔案, 其座標數字是 A4 紙張之大小, 而非圖形實際大小。

欲將 PS 檔案轉換為 EPS, 請利用 cwTeX 所提供之 makeeps 批次檔。此一批次檔事實上是由幾個程式組合而成, 包括: Roland Bless 所寫的 ps2eps , Glyph & Cog 公司的 pdftops , 以及 MiKTeX 系統內之 epstopdf 。其中, ps2eps 須使用 perl 程式。以上過程看似複雜, 但執行方法簡單。更重要的是, 經過測試, 此一方法能產生正確規格的 EPS。除了 EPS 檔案外, 此批次程式同時產生 PDF 圖形檔, 若是以 pdflatex 排版, 文稿內可引入 PDF 圖形檔。

現假設由 PS 印表機驅動程式所產生之檔案已重新命名為 PStest.ps , 並儲存於 c:\xtemp 。請進入 DOS, 再進入 c:\xtemp , 執行:

```
c:\xtemp>makeeps PStest
```

請注意, 不須鍵入附加檔名。硬碟內將產生 PStest.eps 與 PStest.pdf 兩檔案。

圖 11.4 的例子是選取 Excel 試算表內的一個小方塊, 其內含表格與數字, 以 PS 印表機驅動程式列印出 PS 檔案, 再以 makeeps 輸出為 EPS。如果是要輸出 Excel 之圖形, 畫圖時最好把圖形置於獨立的一頁中, 輸出圖形較為方便。

萃豐庄	田甲數	園甲數			
內海墘庄	110	121	番婆汶	132	53
二重溪	90	86	下陰影窩	170	121
石磊仔	218	92	伯公崗	260	291
坑尾	99	104	後庄	227	62
白沙墩	367	106	笨仔港	166	72
新城庄	69	10	十五間	287	54
田心仔	125	76	蚵殼港	123	140
下田心仔	370	153	頭湖	51	158
新屋	237	121	大坡庄	225	104

圖 11.4: 由 Excel 所產生之 EPS 圖形

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。若使用 Corel Draw 繪圖軟體, 我們直接在版面上拉出線條、上色彩、加文字, 儲存之後還可以將圖檔另外輸出為 EPS 格式之圖檔。一般的科學軟體, 如 Mathematica 或 Gauss 等, 大都可以輸出 PostScript 檔案。一旦取得 PS 檔案, 即可進一步以 makeeps 輸出 EPS。底下將以幾個軟體為例, 說明轉換檔案時須注意之細節。

相反的, 若使用 Windows 之 Excel 試算表軟體, 我們通常是先輸入數字, 再畫出資料圖 (chart)。Excel 並無內建輸出 PS 圖檔之功能, 但我們可以利用 PS 印表機驅動程式將圖形輸出為 PostScript 檔案。另外, 一個值得一提的軟體是 Visio。我們並無使用此軟體的經驗, 但討論區經常提及。依據使用者所述, 較新版的 Visio 無法直接輸出 EPS, 因此也必須經由上一小節所述之 PostScript 驅動程式輸出。

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, 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。以下所介紹的主要是我們所熟悉使用者。如果你使用不同版本, 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歐美各國之業者所開發, 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, 讓我們可以在圖形中鍵入中文字, 但是各軟體對於中英文之處理並不相同。譬如說, 有些軟體會把圖形所鍵入之中文字轉成描點圖形。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是儲存為描邊字形。

以上兩種處理方法的主要差別是在於列印品質的不同。如果你注重品質，鍵入英文與數字時應選用 Windows 之英文字型。在 Windows 下，我們可以選用之英文字型很多。若無特別偏好，請選擇較常見之字體，如 Times New Roman 或 Arial。另外，如上一節所述，如果要使用 \psfrag 指令替代字元，我們也須選用 Windows 之英文字型。

繪圖軟體: Corel Draw

以 Corel Draw (9.0 版) 畫好圖形，存檔之後，即可依以下步驟輸出 (export) EPS 格式之檔案。

File > Export

輸入檔名之後，接下來的畫面有幾個選項，其中 Image header 請勿勾選；而 Export Text 應選擇 Text，並勾選 Include fonts。

試算表軟體: Excel

Excel 所畫之資料圖可以利用 PostScript 印表機驅動程式輸出為 PS 檔案，再利用 makeps 轉為 EPS。不過，Excel 之畫圖功能有限，如果你對其所畫圖形之品質不滿意，在畫好圖之後，可利用 Windows 之複製與貼下 (Copy and Paste) 功能，將圖形貼到 Corel Draw 中，經過修改再輸出為 EPS。請特別注意，自 Corel Draw 貼下時，須選用 Paste special 項下之 Picture (Metafile)。

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,0,2π},{y,0,2π}]
```

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。

不過，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，圖形變成很小，或者版面一閃而過。解決辦法是把附加檔名改為 .ps，再以上一節介紹的 makeps 轉換為 EPS。圖 11.5 是排版結果。

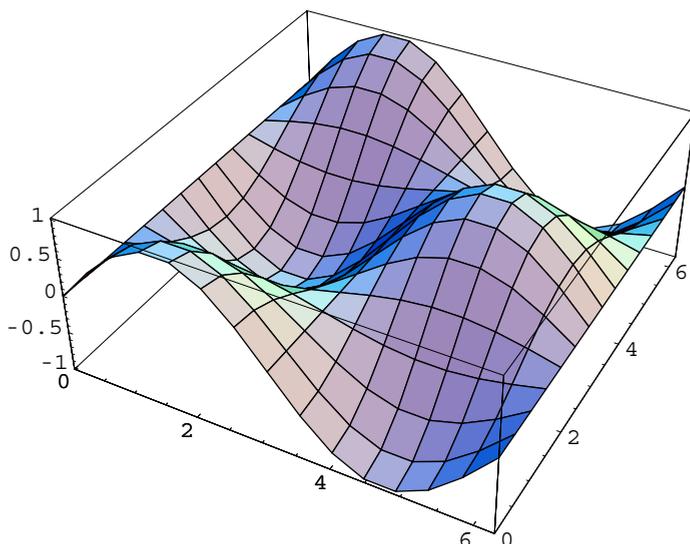


圖 11.5: 引用 Mathematica 軟體所產生的 EPS 圖形

11.4 引用描點圖形

描點圖形格式衆多，最常用的包括 JPEG 與 PNG 等，例如，數位相機照片通常儲存為 JPEG 格式。描點圖形可由 `latex + dvips` 排出，也可由 `pdflatex` 排出。舉例來說，圖 11.6 是一張數位照片，檔名為 `cx3-picnic.jpg`。如果是用 `pdflatex` 排版，則使用下列指令即可引入描點圖形：

```
\includegraphics[width=0.8\textwidth]{cx3-picnic.jpg}
```

`pdflatex` 本身就具備處理描點圖形的能力，引入描點圖形很簡單。但是，`latex + dvips` 處理描點圖形之能力較弱，以此方法排版描點圖形稍微麻煩一些。

上一節說明，PostScript 是一描邊圖形語言，但事實上它也可以處理描點圖形。如果使用 `latex + dvips` 排版，則引入描點圖形最方便的方法是先把圖形檔轉換為 EPS 格式。¹ Dirk Krause 提供 `bmeps` 軟體，可將描點圖形 (JPEG 與 PNG 格式) 即時轉換為 EPS，使用起來很方便。

`bmeps` 軟體是經由 `dvips` 程式作即時轉換。換言之，文稿先以 `latex` 排版，再以 `dvips` 輸出為 PS 檔案時，若其中引入描點圖形，`bmeps` 可將之即時轉換為 EPS，再引入文稿內。雖然 `bmeps` 可作即時轉換，但 `latex` 排版時仍然需要知道圖形之尺寸，才能在版面上留出排版圖形之空白。圖形尺寸是以 `bounding box`

¹感謝余家銘先生對此部分提供協助。



圖 11.6: 野餐

來衡量。所謂 bounding box, 是指包住圖形之長方形外框。如果知道 bounding box 之座標, 我們就知道圖形的大小。

那麼, bounding box 之座標如何找出? 最簡單的方法是利用 `bmeps` 程式來計算。圖形檔 `cx3-picnic.jpg` 置於 `c:\xtemp` 檔案夾內, 進入 DOS 視窗, 再進入 `c:\xtemp` 檔案夾, 執行:

```
c:\xtemp>bmeps -b cx3-picnic.jpg cx3-picnic.bb
```

圖形之 bounding box 座標即算出, 並儲存於 `cx3-picnic.bb` 檔案內。排版時, `latex` 會自動取用檔案內儲存之 bounding box 的座標值, 並在版面上留出適當的空白以置於圖形。

如果你有許多的 `.png` 與 `.jpg` 描點圖形, 要一個一個輸出 `.bb` 檔案也不方便。此時, 可使用 `makebb` 批次檔。如果描點圖形檔置於 `d:\tex\graph` 檔案夾內, 請進入 DOS, 再進入 `d:\tex\graph`, 鍵入以下指令:

```
d:\tex\graph>makebb
```

即可輸出所有描點圖形的 `.bb` 檔案。

11.5 引用 PDF 圖形

若圖形檔是 PDF 格式, 要引入文稿內必須以 `pdflatex` 排版, 或者以 `latex` 排版, 但以 `dvipdfmx` 輸出 PDF。如果是 `dvipdfmx` 輸出 PDF, 則引用 `graphicx` 巨集套件須加入驅動程式選項:

```
\usepackage[dvipdfm]{graphicx}
```

排版之前，我們須先找出 PDF 圖檔之 bounding box 座標，dvipdfm 程式所提供的 ebb 程式即此用途。假設圖形檔名為 PDFtest.pdf，置於 c:\xtemp 檔案夾內，進入 DOS，再進入 c:\xtemp，執行：

```
c:\xtemp>ebb PDFtest.pdf
```

即產生 PDFtest.bb，內含 bounding box 之座標。

反之，如果是以 pdflatex 排版，程式可自行算出 PDF 圖形檔之 bounding box。除了 PDF 圖形檔之外，pdflatex 也可以處理 JPEG 與 PNG 等描點圖形檔案，不過，它無法處理 EPS 圖形檔。如果你習慣以 pdflatex 排版，但圖形檔卻是 EPS 格式，解決的辦法是以 MiKTeX 內附之 epstopdf 先將圖形轉換為 PDF。例如，若檔名為 test.eps，置於 c:\xtemp，執行以下指令即產生 test.pdf：

```
c:\xtemp>epstopdf test.eps
```

以上的辦法雖然可將 EPS 圖形檔轉換出 PDF，但如果文稿內使用 \psfrag 指令替換 EPS 圖形之字元，這個辦法還是沒有解決問題。原因是，pdflatex 程式無法處理 \psfrag 指令。簡單歸納我們的問題如下：

- 以 pdflatex 排版，
- 欲使用 \psfrag 指令替換 EPS 圖形檔內之字元。

欲解決以上問題，我們可使用 ps4pdf 巨集套件（作者 Rolf Niepraschk）。

ps4pdf 巨集套件之運作方式可簡單說明如下。將前面圖 11.3 之例子略作更改如下：

```
\usepackage{graphicx,psfrag,ps4pdf}
\PSforPDF{\usepackage{graphicx,psfrag}}
...
\begin{figure}
\centering
\PSforPDF{
\psfrag{J}{\small 在來米}
\psfrag{C}{\small 甘蔗}
\psfrag{P}{\parbox{.4\textwidth}{

$$P(t)=\beta_0+\beta_1\frac{R_{R1}(t)}{P_C(t)}$$

}}
\includegraphics[width=0.9\textwidth]{cx3-ricePS.eps}
\end{figure}
```

首先，全文設定區須引用 `ps4pdf` 巨集套件，並加入一行 `\PSforPDF` 指令之設定。文稿內仍依標準指令引入圖形，但是，`\includegraphics` 指令以及其上所有的 `\psfrag` 指令必須置於 `\PSforPDF{...}` 指令內。

若文稿檔案名為 `test.ctx`，則執行 `cwtextex` 與 `pdflatex` 之後，接下來進入 DOS，再進入 `c:\xtemp`，執行：

```
c:\xtemp>gpdf test
```

其中，`gpdf.bat` 為一批次檔，內容為：

```
latex %1
dvips -Ppdf -o %1-pics.ps %1.dvi
ps2pdf %1-pics.ps %1-pics.pdf
```

以上第一次執行 `pdflatex` 時，`ps4pdf` 巨集套件將 `\PSforPDF{...}` 之內容存為一暫存檔，其中含有引用圖形之指令，以及 `\psfrag` 指令。執行 `gpdf.bat` 批次檔時，電腦將啟動 `latex` 排版此一暫存檔，其中之英文字元已經過替換。排版之結果先以 `dvips` 轉換為 PS 檔，再以 `ps2pdf` 轉換為 PDF。

以上所產生之 PDF 圖形檔案內，字元已經過轉換，故可引入文稿內。最後，我們須再一次執行 `pdflatex` 即可得到正確的結果。後續的排版中，除非圖形有所修改或增加，否則不須再執行 `gpdf.bat`。

如果文稿內引用了 5 個 EPS 圖形，`\PSforPDF` 指令會將這 5 個圖形全部納入在一個 PDF 檔，取名為 `test-pics.pdf`。此檔共有 5 頁，每一頁就是一個圖形。若想把各圖形分離出來，可以 `GSview` 讀取 `test-pics.pdf`，利用 File 表單下之 `Convert`，即可將各圖形分別儲存。

12 圖形與彩色

本章主要說明如何在 \TeX 文稿中自行畫出圖形以及如何上顏色。第 11 章介紹引用外製圖形之指令時，曾說明如何將圖形放大或縮小，這是透過 `graphicx` 巨集套件之指令。此一巨集套件也提供旋轉與縮放文字圖表之指令。彩色的功能主要是透過 `color` 巨集套件。我們可將圖形、文字上色，或加上灰階 (grayscale) 之背景方塊。 \TeX 提供 `graphicx` 及 `graphics` 兩套處理圖形之巨集套件。兩者之功能相近，但前者使用較方便，故本章僅介紹前者。

除了引用外製圖形之外， \TeX 提供 `picture` 指令環境，可畫直線與曲線。雖然功能相對簡單，但如果我們所畫的圖形並不複雜，`picture` 指令環境是方便的工具，12.4 節將略作說明。

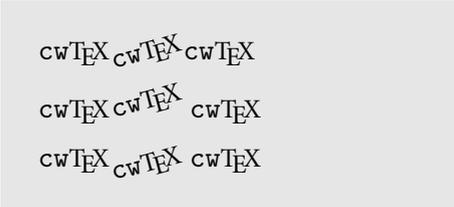
12.1 旋轉與縮放文字圖表

\TeX 的繪圖功能有限，大部分的繪圖功能必須透過預覽與列印軟體來執行。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。等到預覽與列印時，圖形才由軟體引進版面上。使用圖形與彩色之巨集套件，排版文稿時須以 `dvips` 程式轉換為 PostScript 檔案，再以 `GSview` 預覽/列印。若直接以 `YAP` 預覽/列印，有時候無法得到正確的結果。或者，我們也可以用 `pdflatex` 排版為 PDF 檔案，再以 `Reader` 預覽/列印。

因為圖形與彩色是透過預覽/列印軟體呈現，因此，使用 `graphicx` 巨集套件或 `color` 巨集套件時，我們須指定使用那一個驅動程式預覽與列印。最常用之驅動程式是 `dvips`，`pdftex` 與 `dvipdfm`。如果使用前兩者之一，排版程式會自動判斷，不須再加入設定選項。反之，如果是使用 `dvipdfm` 或 `dvipdfmx` 輸出 PDF，文稿內引用巨集套件時須加入驅動程式選項：

```
\usepackage[dvipdfm]{graphicx}
```

同理，使用 `color` 巨集套件時，也須作類似之設定。



```

\usepackage{graphicx}
\newcommand{\cw}{\texttt{cw}\TeX}
...
\cw \rotatebox[origin=c]{15}{\cw}%
    \cw \\\[2pt]
\cw \rotatebox{15}{\cw} \cw\\\[5pt]
\cw \rotatebox[x=20pt,y=10pt]{15}%
    {\cw} \cw

```

圖 12.1: 旋轉文字圖表

12.1.1 旋轉文字圖表

排版時，我們有時候須旋轉某部分之文字圖表。譬如，若表格橫寬太大，必須將之旋轉 90 度才能排入版面。要把文字圖表旋轉某一角度，可使用 `graphicx` 之 `\rotatebox` 指令，指令格式如下：

```

\rotatebox[options]{angle}{material}

```

其中，*options* 代表選項，*angle* 為旋轉角度，*material* 代表欲旋轉之文字圖表。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，全部文字須置於一個段落內。換言之，欲旋轉之文字或表格中，不得空一行或加上 `\par` 指令，但是可使用換行指令 `\\`。

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 說明不同旋轉軸心點之效果。本例中，我們先定義排版 `cwTeX` 標識符號之指令。為了方便比較，每一行排出三個標識符號；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，因此以左下角之基準點為軸心；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

$\text{cwt}_{\text{E}}\text{X}$
 $\text{cwt}_{\text{E}}\text{X}$

```
\usepackage{graphicx}
\newcommand{\cw}{\texttt{cw}\TeX}
...
\scalebox{2}{\cw} \[4pt]
\scalebox{1.2}[1.4]{\cw}
```

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}{...}` 指令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尺寸，如 2 公分，而高度希望同比例放大，可以使用下列指令：`\resizebox*{2cm}{!}`，請參見底下的例子。

$\text{cwt}_{\text{E}}\text{X}$

```
\usepackage{graphicx}
\newcommand{\cw}{\texttt{cw}\TeX}
...
\resizebox*{2cm}{!}{\cw}
```

`\scalebox` 指令之水平倍數若為負值，結果為鏡面反映。此外，`\reflectbox` 指令也可以排版鏡面反映之效果。

$X_{\text{q}}\Gamma\omega$
 $X_{\text{q}}\Gamma\omega$

```
\usepackage{graphicx}
\newcommand{\cw}{\texttt{cw}\TeX}
...
\reflectbox{\cw} \[2pt]
\scalebox{-1}[1]{\cw}
```

12.2 彩色圖文

欲將文字或圖形加上色彩，最簡單的方法是使用 $\text{E}_{\text{T}}\text{X}$ 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)。

如 12.1 節 (頁 213) 之說明， $\text{E}_{\text{T}}\text{X}$ 是透過預覽/列印程式呈現圖形與彩色，因此，使用 `color` 巨集套件時，必須配合以特定之預覽/列印程式。最常用的方法是以 `dvips` 程式產生 PostScript 檔案，再以 `GSview` 預覽/列印。或者，我們也可以使用 `pdflatex` 排版，以 `Reader` 預覽/列印。若是以 `YAP` 預覽，有些色彩可能無法正確顯示。

最常見之色彩應用有兩類，一是將文字著色，一是在文字或圖表之後加上灰階或顏色背景。就色彩之選擇而言，`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 驅動程式。如果我們是使用 dvips 或者 pdflatex 排版, 排版程式可以自動判斷, 故以上之 dvips 事實上可省略不加。不過, 如果是使用 dvipdfm 或 dvipdfmx 輸出 PDF, 則必須加入 dvipdfm 選項。

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。另外一種下指令的方式如下:

```
{\color{red}{\ctxfbb 請注意}}, 三分鐘之後 ...
```

從 `\color{red}` 指令開始, 以下之文字皆以紅色排出。因此, 我們須以左右大括號限定色彩改變之範圍。

以上以 red 指令選擇紅色。L^AT_EX 可以直接選用之色彩選項為 black, white, red, green, blue, yellow, cyan 與 magenta。若要自行設定顏色, 請見下一小節之說明。

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, 首先須以 `\definecolor` 指令定義灰階色之深淺度。灰階色深淺是以一介於 0 與 1 之間的數字代表, 1 表示全白, 0 表示全黑。本例定義 slight 為 0.75 度之灰階色, 再以之排版灰階方塊:



```
\usepackage{color}
\definecolor{slight}{gray}{0.75}
\fbboxsep=15pt
...
\colorbox{slight}{\Large 灰色背景。}
```

第 2 行指令定義灰階之深淺, gray 是設定深淺度之指令, slight 是自行選定之灰階色名稱。本例之深淺度為 0.75。另外, 背景外框與文字之距離可由 `\fbboxsep` 指令控制, 本例設定為 15pt。

若要將整段文字加上灰階背景, 可將段落文字排版於 `minipage` 指令環境內, 再以顏色指令上色。例如:

```
\colorbox{slight}{\begin{minipage}
...
\end{minipage}}
```

如果要將整頁版面加上背景顏色, 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` 指令在其中排版出白字:

請特別注意!

```
\usepackage{color}
\definecolor{dark}{gray}{0.4}
\fbboxsep=10pt
...
\colorbox{dark}{\textcolor{white}%
{\LARGE \ctxfbb 請特別注意 !}}
```

指令第 5 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。

12.2.1 選擇色彩

上一小節說明如何對文字或圖形上顏色, 色彩是以特定之名字選用, 如 `red` 或 `blue`。如果要選擇其他顏色, 可以依特定之色彩模式 (`color model`) 自行定義。如果覺得自行定義太麻煩, 我們也可以使用現成之定義。

Uwe Kern 所寫的 `xcolor` 巨集套件其中提供許多成之色彩定義, 列於表 12.1 以供參考。表中前 3 列 (共 15 種顏色) 為 `xcolor` 內定之顏色, 只要引入 `xcolor` 巨集套件即可使用。第 4 列開始之顏色 (若不屬於以上 15 種之內), 引用 `xcolor` 巨集套件時須加入 `dvipsnames` 選項才能選用顏色:

```
\usepackage[dvipsnames]{xcolor}
```

實際之色彩請見 `xcolor` 之說明檔, 置於 `c:\texmf\doc\latex\xcolor` 子目錄內。

如果以上之色彩不符合你的需求, 我們也可以使用色彩模式 (`color model`) 自行定義色彩。目前, 常用的色彩模式包括 `CMYK`, `RGB` 等。以 `CMYK` 為例, 任何色彩是由 `Cyan`, `Magenta`, `Yellow`, `blacK` 等 4 個基本顏色組成, 每一種基本顏色之值介於 0 與 1 之間。因此, 我們可自行定義一色彩名為 `classic` 如下:

表 12.1: 以 xcolor 巨集套件選用色彩

black	blue	brown	cyan	darkgray
gray	green	lightgray	magenta	orange
purple	red	violet	white	yellow
Apricot	Aquamarine	Bittersweet	Black	Blue
BlueGreen	BlueViolet	BrickRed	Brown	BurntOrange
CadetBlue	CarnationPink	Cerulean	CornflowerBlue	Cyan
Dandelion	DarkOrchid	Emerald	ForestGreen	Fuchsia
Goldenrod	Gray	Green	GreenYellow	JungleGreen
Lavender	LimeGreen	Magenta	Mahogany	Maroon
Melon	MidnightBlue	Mulberry	NavyBlue	OliveGreen
Orange	OrangeRed	Orchid	Peach	Periwinkle
PineGreen	Plum	ProcessBlue	Purple	RawSienna
Red	RedOrange	RedViolet	Rhodamine	RoyalBlue
RoyalPurple	RubineRed	Salmon	SeaGreen	Sepia
SkyBlue	SpringGreen	Tan	TealBlue	Thistle
Turquoise	Violet	VioletRed	White	WildStrawberry
Yellow	YellowGreen	YellowOrange		

前 3 列 (15 種顏色) 為 xcolor 內定之顏色; 第 4 列開始之顏色 (若不屬於以上 15 種之內), 引用 xcolor 巨集套件時須加入 dvipsnames 選項才能選用。

```
\usepackage{color}
\definecolor{classic}{cmyk}{1.0,0.6,0,0.45}
```

經過以上定義之後, 文稿內即可以 classic 色彩排版。關於色彩之使用, 除了上述 xcolor 巨集套件之說明外, 亦請參考 color 巨集套件之說明, 以及 [Goossens et al. \(1997\)](#)。

12.3 圖形小工具

本節介紹幾個與圖形有關的小工具套件, 有助於版面設計。其中, `lettrine` 巨集套件用於排版所謂的 drop caps, `wrapfig` 巨集套件則用於排版面邊沿的小圖形。

12.3.1 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, 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。

除了設定行數之外，我們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`則指示要將單字放大到使其高度高於正常值的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。下一小節第一個放大字使用 `\textcolor` 指令設定以40%灰階排版。排版指令如下：

```
\usepackage{color,lettrine}
\definecolor{heavy}{gray}{0.4}
...
\lettrine[lines=2,lhang=0.2]{\textcolor{heavy}{\ctxfbb 引}}{ } 用 ...
```

12.3.2 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文字段落的左邊或右邊。甚至是，圖表置於方塊內，四周以文字包圍。欲排版此種版面，可使用 Donald Arsenau 所寫的 `wrapfig` 巨集套件。舉例言之，下一段文字旁邊的圖形是以 `wrapfig` 指令環境引入，指令如下：

```
\intextsep=0pt
\begin{wrapfigure}[5]{l}[1cm]{3.4cm}
\includegraphics[width=3cm]{cat.eps}
\end{wrapfigure}
\texttt{wrapfig} 巨集套件所提供之指令環境 ...
```

以上之指令是下在本段文字之後，指令之後立即接上文字。



`wrapfig` 巨集套件所提供之指令環境取名為 `wrapfig`。指令環境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。本例指令第2行之第1選項5，設定圖形高度占5行文字。若不加入此一選項，巨集套件會自動計算。第2選項若為 `l` 指示將圖形置於版面左邊，反之，選項若為 `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公分。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。

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。本例指令第一行為 `\intertextsep=0pt`,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,請見其說明檔。作者將使用說明直接置於 `wrapfig.sty` 檔案末端,檔案儲存於 `c:\texmf\tex\latex\wrapfig` 子目錄內。

12.3.3 contour 巨集套件

大部分之商業繪圖軟體,如 `Corel Draw` 或 `PhotoShop`,都可以創造立體陰影 (dropping shadows) 效果。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比不上商業繪圖軟體,但使用起來很方便。

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,color}
\definecolor{cnheavy}{gray}{0.2}
\definecolor{cnlight}{gray}{0.75}
...
\fbboxsep=15pt
\colorbox{cnlight}{\contour{white}{\color{cnheavy}
\fontsize{40}{50pt}\selectfont 台大經濟系}}
```

排版結果為:



其中, `cnheavy` 為 0.2 灰階度。若改為其他深淺度之灰階色,排版效果會不一樣,有興趣者請自行嘗試。

12.4 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,我們也可以透過巨集套件引用 `PostScript` 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。

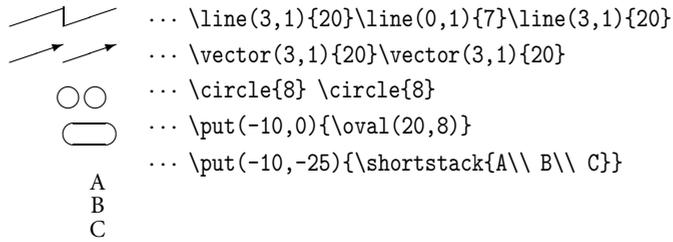


圖 12.3: picture 指令環境

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 之長方形，可定為左下角座標為 $(0,0)$ ，`{\MaQ\cH172}\z{\MaQ\cH44}\z{\McQ\cH105}\z{\MbQ\cH209}` 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}
```

即選定公厘為長度單位。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（寬度為 10 公分，高度為 6 公分），其左下角之位置與原基準點之距離為 $(20\text{mm}, -10\text{mm})$ ：

```
\setlength{\unitlength}{mm}
\begin{picture}(100,60)(20,-10)
...
\end{picture}
```

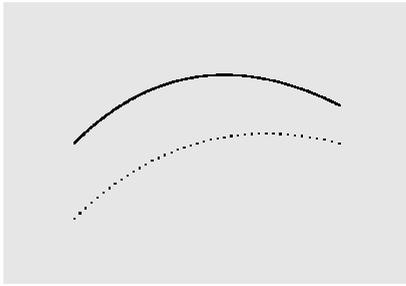
在 `picture` 指令環境內，畫直線之指令為 `\line`。譬如 `\line(2,3){10}` 指令將畫一直線，起始點為 $(0,0)$ ，`{\MbQ\cH106}\z{\MaQ\cH178}\z{\MbQ\cH209}` $(2,3)$ ；長度為 10。進一步利用 `\put` 指令，即可將此直線移至版面上選定的位置。畫箭號之指令為 `\vector`，圓圈之指令為 `\circle`，橢圓之指令為 `\oval`。圖 12.3 列出幾個簡單的例子。我們將 `picture` 指令環境內之畫圖指令簡單歸納如下：

- `\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： $\bullet \bullet \bullet$

- `\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*。參見圖 12.3 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。
- `\shortstack`: 將文字圖表垂直疊在一起, 各行之間須以斷行指令隔開。
- `\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在框內靠上, 靠下方, 靠左, 靠右。以上之選項可進一步組合, 譬如 [tl] 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 `\qbezier` 指令可畫出曲線圖, 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, 但讓使用者能更方便地繪出圖形。譬如, Peter Vanroose 之 `trees` 巨集套件用於繪製樹狀圖甚為方便。另外一些巨集套件



```
\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

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。有意使用者, 請參考 [Goossens et al. \(1997\)](#)。

12.5 PSTricks 巨集套件

上一節簡單介紹如何使用 `picture` 指令環境在文稿內直接繪製線條圖形。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。不過, 除非你非常熟悉 PostScript 指令, 否則直接鍵入 PostScript 指令是不可能的。

要在文稿中內引入 PostScript 畫圖指令, 可行的方法是透過應用程式與巨集套件。有名的應用程式之一是 `MetaPost`, 作者 John Hobby。應用此一程式, 我們可以輸入 `MetaFont` 指令, 程式將輸出 PostScript 畫圖指令以引入排版文稿內。欲使用此一程式, 我們須熟悉 `MetaFont` 指令。

另一組有名的巨集套件是 Timothy van Zandt 所寫的 `PSTricks`。此巨集套件之畫圖功能甚強, 指令也相當完整者。[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。`PSTricks` 巨集套件提供許多指令, 我們無法在此一一說明。本節僅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, 因此我們必須以 `latex + dvips` 排版。

`PSTricks` 早在 1993 年就發展出來, 它和後來的新版 \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，排版時會出現錯誤。

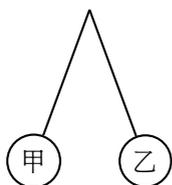
整套 PSTricks 巨集套件包含幾個較小的巨集套件，各有不同功能，底下僅以幾個例子展示其功能。第一個例子是將文字加入 PS 指令所繪成的橢圓中。指令中設定橢圓是以紅線畫出。我們還可以進一步使用指令讓 `cwTeX` 標識符號以另一種顏色排版出來。若使用 GSview 軟體預視，即可看到彩色的圖案；



```
\usepackage{pstcol,pst-grad}
\psovalbox[linecolor=red]
{\Large{\tt cw}\TeX}
```

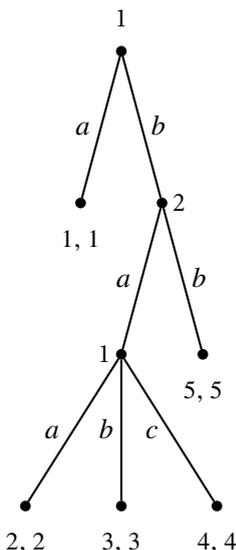
其次，我們說明如何繪製樹狀圖 (trees)。許多學科領域之研究論文都須繪製樹狀圖，但在 `TeX` 文稿內直接繪製樹狀圖並不容易。PSTricks 繪製樹狀圖之能力甚佳，如果你經常須繪製樹狀圖，則花一點時間了解 PSTricks 之畫圖指令，絕對是一個值得的投資。

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，其上下共有 3 層。



```
\usepackage{pst-node,pst-tree}
\psset{labelsep=2pt,tnpos=a,radius=2pt}
\pstree{\TC*~{1}}
{\TC*~{~[tnpos=b]{$1,1$}\tlput{$a$}
\pstree{\TC*~{~[tnpos=r]{2}\trput{$b$}}
{\pstree{\TC*~{~[tnpos=l]{1}\tlput{$a$}}
{\TC*~{~[tnpos=b]{$2,2$}\tlput{$a$}
\TC*~{~[tnpos=b]{$3,3$}\tlput{$b$}
\TC*~{~[tnpos=b]{$4,4$}\tlput{$c$}
}
\TC*~{~[tnpos=b]{$5,5$}\trput{$b$}
}
}
```


13 投影片與信函

如果你是學生，某些課程會要求你上台報告。如果你在學校教書，授課與發表論文是固定的工作。出考題與提供解答也是家常便飯。在作口頭報告或演講時，單槍投影機幾乎已經是不可或缺的工具。如果你所提交的書面報告是以 \LaTeX 排版，則投影片內容順理成章也應該是以 \LaTeX 排版。 \LaTeX 主要是用來排版書籍與論文，但是它也可以用來處理日常文書事務，如排版信函與大宗郵件，或者出考題與提供解答。本章第1節即介紹如何使用 `beamer` 巨集套件排版投影片。2–4節說明如何排版信函，第5節則介紹排版考題與解答之巨集套件。

13.1 投影片

單槍液晶投影機發明之後，目前幾乎已完全取代傳統投影機。單槍投影機的功能是將電腦顯示器上之簡報內容放映出來。在 Windows 作業系統上，Microsoft 的 Power Point 是一套有名的簡報軟體。在 \LaTeX 系統下，我們也可以製作出類似 Power Point 功能之投影片。第2章例2.6 (頁34) 即是以 \LaTeX 排版的投影片。排版結果輸出為 PDF 格式，再以 Reader 播放。

13.1.1 `beamer` 巨集套件

\LaTeX 系統有好幾套製作單槍投影片之巨集套件，其中，`beamer` 是相當受歡迎的一套，作者為 Till Tantau。投影片與一般文稿不同。首先，一般文稿是縱向排版，投影片則常是橫向排版。為了提高簡報之效果，投影片通常以彩色呈現，有時候還須加上動畫效果。以 `beamer` 巨集套件排版投影片，除了可以使用 \LaTeX 之排版指令外，我們還可以使用一些特別的指令以創造上面所述的效果。

本節將把第2章的例2.6略作擴充，仔細說明各指令之功能。例子之內容請見圖13.1，檔案取名為 `ex-bm.ctx`，存於 `c:\texmf\cwtex\examples\beamer` 子目錄內，供自行測試。排版一般文稿時，我們通常使用 `article` 或者 `book` 文稿類別，若以 `beamer` 排版投影片則須使用 `beamer` 文稿類別。本例的第1行為

```

1 \documentclass[14pt,dvips,%
2   hyperref={bookmarks=false}]{beamer}
3
4 \usetheme{Malmoe}
5 \usecolortheme{dolphin}
6 \usefonttheme{professionalfonts}
7 \useoutertheme{miniframes} %{\infolines}
8 \usepackage{xmpmulti}
9
10 \linespread{1.2}
11
12 \newenvironment{num}
13   {\leftmargini=6mm\leftmarginii=8mm
14   \begin{itemize}}{\end{itemize}}
15
16 \ctxfr
17 \title{經濟學是什麼?}
18 \author{吳聰敏\ 台大經濟系}
19 \date{2005.08}
20 \begin{document}
21 \begin{frame}
22 \titlepage
23 \end{frame}
24
25 \raggedright
26 \begin{frame}
27 \normalsize
28 \tableofcontents
29 \end{frame}
30
31 \section{矮人頭骨與登山小屋}
32 \subsection{Flores 矮人}
33
34 \begin{frame}
35 \frametitle{Flores 矮人頭骨}
36 \includegraphics[width=.8\textwidth]{
37   {cx3-Flores.jpg}\label{Flores}
38 \end{frame}
39
40 \section{有味道的廁所}
41 \begin{frame}
42 \frametitle{學生為何要掃廁所?}
43 \begin{num}
44 \item 台北最乾淨的廁所? \pause
45 \item 台北味道最重 (最不乾淨) 的廁所? \pause
46 \item 學生為何要掃學校的廁所? \pause
47 \item 必須掃\alert{老師的廁所?}
48 \end{num}
49 \end{frame}
50
51 \begin{frame}
52 \frametitle{Flores 矮人}
53 \begin{num} % Do not add \pause
54 \item<1-> 大頭的好處 (與壞處)?
55 \item<2-> 好處1: 對付敵人
56 \item<3-> 好處2: 吸引異性
57 \item<1-> 壞處: \alert{頭太大, 餓死的機率高}
58 \end{num}
59 \uncover<4->{\hyperlink{Flores}{\color{violet}
60   Flores island}: 無大型獵食動物}
61 \end{frame}
62
63 \begin{frame}
64 \frametitle{供給與需求}
65 \includegraphics<1>[width=.8\textwidth]{
66   {ex-bm-0.png}
67 \includegraphics<2>[width=.8\textwidth]{
68   {ex-bm-1.png}
69 \includegraphics<3>[width=.8\textwidth]{
70   {ex-bm-2.png}
71 \end{frame}
72 \end{document}

```

圖 13.1: beamer 文稿類別

\documentclass 指令, 但使用 beamer 文稿類別。我們可以由 latex + dvips 排版, 再以 ps2pdf 程式輸出 PDF; 也可以使用 pdflatex 排版。以 pdflatex 排版較為直接, 但如果文稿內引用的 EPS 圖形, 而且使用 psfrag 巨集套件轉換字元, 應以 latex + dvips 排版, 因為 pdflatex 程式不容易處理 (參見 11.5 節之說明)。

投影片裡, 我們仍可使用 \section 與 \subsection 指令排版節與小節標題。排版之後, 節與小節標題會出現在投影片上方之工作列上。演講時, 若播放某一張投影片, 該節 (或小節) 之標題會明顯反白, 這有助於講者掌握演講節奏。此外, 若投影片內容有 4 節, 目前你已講到第 3 節, 但第 3 節的內容與第 1 節有關。直接點選投影片上方之第 1 節之標題 (或代表小節標題之符號), 即可連接到該節。



超連結

如果投影片內要有上述超連結之效果, 則 `\documentclass` 指令內必須加入 `dvips` 或者 `pdftex` 選項。若不加選項, 仍可跑出 PDF 檔案, 但不會有超連結之效果。節與小節之標題可排版於目錄中, 26–29 行指令之作用即將目錄頁排版於一張投影片內。我們特別加上 `\normalsize` 指令指示使用標準字級排版。若不加此指令, \TeX 會選用較大之字體。第 21–23 行之指令則排版出題目頁。

投影片的字體不能太小, 本例選用 14pt 字體排版。我們也可以用 17pt 或更大的 20pt。第 4–7 行為 `beamer` 為設計投影片版面之指令。第 8 行引入 `xmptmulti` 巨集套件, 目的是爲了要以動態方式呈現圖片。以上指令之細節稍後再作說明。中文稿之行距須比英文稿之內定值大, 第 10 行以 `\linespread{1.2}` 指令將行距加大爲內定值的 1.2 倍。

每一套投影片有個主題, 題目以 `\title` 指令排版, 作者以 `\author` 排版, 日期以 `\date` 排版, 以上三個指令是標準的 \TeX 指令。第 21–23 行指令是將題目、作者與日期排入一張投影片內。題目與作者名字會排版於投影片頁足。如果題目太長, 頁足之空間可能不夠, 此時我們可以加入簡短題目之選項, 例如:

```
\title[經濟學]{經濟學是什麼?}
```

投影片之內容主要是條列重點, 故排版時經常使用 `itemize` 或 `enumerate` 指令環境。本例第 12–14 行以 `itemize` 指令環境爲基礎, 把條列項目之格式略作調整, 重新定義一指令環境, 取名爲 `num`。主要的改變經由 `\leftmargini` 之設定, 調整各條列項目之內縮距離。

13.1.2 動態呈現

每一張投影片之指令與文字須置於 `frame` 指令環境, 或者 `\frame` 指令內, 其上可以 `\frametitle` 指令排版標題。若暫且不計入標題與目錄, 第 34–38 行是第一張投影片, 內容是一張圖片, 標題爲「Flores 矮人」。41–49 行是另一張投影片, 此張投影片以 `num` 指令環境條列 4 個項目。這 4 個條列項目中, 前 3 個項目末端都加上 `\pause` 指令, 其功能是創造動態呈現之效果。透過此一指令, 以 Reader 播放 PDF 投影片時, 一進入此投影片之畫面, 螢幕上首先會出現第一條列項, 按下 `[Enter]` 之後, 第二項才顯示出來。再按下 `[Enter]`, 畫面上即出現第三項。

`\pause` 指令除了加在條列的項目後面外, 也可以用於一般的文稿內。另外, 第 47 行指令中使用了 `\alert` 指令, 其作用是把「老師的廁所?」一串文字變爲紅色。事實上, 我們也可以使用 `\color` 或者 `\textcolor` 指令將特定之字串加上顏色, 參見第 59 行之指令。

動態呈現之方式還可以進一步控制。假設條列項目計有 4 項, 其中的第 4 項是結論。呈現條列項目時, 我們希望第 1 項與第 4 項同時出現, 接下來再呈現第 2 項與第 3 項。第 53–58 行是一個例子, 方法是在 `\item` 指令之後加入控制選項。第 54 與 57 行的 `\item` 指令之後加上 `<1->`, 指示這兩項是在首張投影片內出現。第 55 行指令中的 `<2->`, 則指示這一項要在下一張投影片內出現。第 56 行的條列項則出現於再下一張的投影片內。

除了 `\pause` 指令之外, 還有其他一些指令也可以表現動態之效果。例如, 第 59 行之 `\uncover<4->` 指令指示此行文字應在第 4 張投影片才出現。請注意, 使用 `<1->` 指令控制時, 條列項之後不須再加入 `\pause` 指令; 而 `\uncover` 指令之內容則須置於大括號內。

第 59–60 行指令內容如下:

```
\hyperlink{Flores}{\color{violet} Flores island}: 無大型...
```



動態呈現

`\hyperlink` 指令之作用類似 \TeX 之 `\ref` 指令。在一般文稿裡, 我們以 `\label` 指令在章節或圖表之標題處下標籤, 再以 `\ref` 指令徵引。本例之標籤位於第 37 行: `\label{Flores}`, 第 59 行之 `\hyperlink` 是徵引指令。排版之後, 當條列項目出現於畫面時, `Flores island` 為紫色。以滑鼠點選, 畫面即跳到指令第 37 行之圖形畫面位置。若要回到第 59 行之畫面, 以滑鼠點選 `Reader` 下方工作列上之左向箭頭即可。

以上之超連結功能是透過 `hyperref` 巨集套件而產生的。`beamer` 巨集套件會自動引入 `hyperref`, 此巨集套件內定的選項之一是啓動 `bookmarks` (書籤) 功能, 亦即在 PDF 畫面左邊顯示文稿之章節標題。因為 `beamer` 本身已提供相同之功能, 故本例第 2 行以 `hyperref={bookmarks=false}` 選項將之關閉。此外, `beamer` 也自動引入 `xcolor` 巨集套件, 故文稿內可以直接使用 `\color{violet}` 指令。

透過 `hyperref` 之功能, 我們也可以從文稿中連結到特定的網址。譬如, 文稿內加入以下指令:

```
\htmladdnormallink{\color{blue}TUG}{http://www.tug.org/}
```

排版之後, 畫面上出現藍色的“TUG”。若電腦連上網路, 以滑鼠點選, 即連結到 \TeX 官方網站。

投影片內可以引入外製圖形, 36–37 行以 `\includegraphics` 指令引入一張 JPEG 點陣圖, 65–70 行則引入 3 張 PNG 格式之點陣圖。後者的指令中, 分別加

入 <1>, <2>, <3>, 目的是要創造動態圖片之效果。第 1 張圖片 `ex-bm-0.png` 畫出經濟學的供給線; 第 2 張圖 `ex-bm-1.png` 在原圖上加入一條需求線; 第 3 張圖 `ex-bm-2.png` 加入另一條需求線。排版之後, 這 3 張圖依序出現時會創造出動態畫圖之效果: 一開始只是一條供給線, 按下 [Enter] 之後, 加入需求線; 再按一次 [Enter], 又加入另一條需求線。

如果有一連串之圖片, 比較簡單的方法是使用 `\multiinclude` 指令, 此一指令是由 `\usepackage{xmpmulti}` 巨集套件所提供, 故我們須自行引入。圖 13.1 例子中之第 63–71 行指令可簡化如下:

```
\usepackage{xmpmulti}
...
\begin{frame}
\frametitle{供給與需求}
\multiinclude[graphics={width=.8\textwidth},format=png]{ex-bm}
\end{frame}
```

選項內之 `format=png` 表示圖檔為 PNG 格式。最右邊大括號內之 `ex-bm` 為圖檔主檔名之第一部分, 實際之圖檔須加入編號, 分別為 `ex-bm-0.png`, `ex-bm-1.png`, `ex-bm-2.png`, ... 等。請注意, 首張之編號為 0。排版時, latex 會搜尋以上之檔案, 依序引入投影片內。引入一張圖形後, L^AT_EX 會自動加入 `\pause` 指令, 故按下 [Enter] 之後, 下一張才出現。

以上 `\multiinclude` 之功能等於是多個依序出現的引入外製圖形之指令, 指令內所加入之選項 `graphics=`, 目的是把 `width=.8\textwidth` 之設定傳給引入外製圖形之 `\includegraphics` 指令, 以將圖形寬度調整為文字版面寬度的 80%。

13.1.3 投影片版面設計

使用 beamer 排版時, 版面之格式有許多選擇。不同之版面格式, beamer 稱之為風格 (theme), 包括版面設計, 字體, 內部 (inner) 風格與外部 (outer) 風格等。每一種風格有其名稱, 4–7 行之指令即是選用特定之風格。其中, `\usetheme` 選擇整體的設計風格, `\usecolortheme` 選擇色彩搭配, 而 `\usefonttheme` 則選擇字型搭配。譬如, 有些色彩風格較強烈, 有些則較冷酷。若不作選擇, beamer 即以內定之設定。

beamer 巨集套件提供許多風格選項, 表 13.1 全部列出以供參考。舉例來說, 圖 13.1 例子指令第 4 行 `\usetheme` 指令所選用之 `Malmoe` 若改為 `Bergen`, 版面之對比會變得比較強烈。同理, 第 5 行之 `dolphin` 若改為 `beetle` 版面會改為藍灰色調。完整之風格圖例及介紹, 請見 beamer 之說明檔。

表 13.1: beamer 風格

指令	風格選項
<code>\usetheme</code>	Antibes, Bergen, Berkeley, Berlin, Boadilla, boxes, Copenhagen, Darmstadt, default, Dresden, Frankfurt, Goettingen, Hannover, Ilmenau, JuanLesPins, Luebeck, Madrid, Malmoe, Marburg, Montpellier, PaloAlto, Pittsburgh, Rochester, Singapore, Szeged, Warsaw
<code>\usecolortheme</code>	albatross, beetle, crane, default, dolphin, dove, fly, lily, orchid, rose, seagull, seahorse, sidebartab, whale
<code>\usefonttheme</code>	default, professionalfonts, serif, structurebold, structureitalicserif, structuresmallcapserif
<code>\useinnertheme</code>	circles, default, inmargin, rectangles, rounded
<code>\useoutertheme</code>	default, infolines, miniframes, shadow, sidebar, smoothbars, smoothtree, split, tree

投影片排版之後須印出來分發給聽眾。要印出投影片之前，請在第 1 行指令中加入 `handout` 選項：

```
\documentclass[14pt,handout,hyperref={bookmarks=false}]{beamer}
```

投影片內若使用 `\pause` 指令，動態呈現個畫面時需要有多張投影片。如果直接列印，我們將會印出多張內容相近之結果。加入 `handout` 選項可以暫時關閉 `\pause` 之功能，以節省列印成本。最後，列印時若要在頁足印出頁碼，第 7 行應改為

```
\useoutertheme{infolines}
```

圖 13.1 例子內容主要是文字，如果有數學式，可利用 \LaTeX 指令排版於投影片內。同理，表格也是依標準指令排版。

如果你要參考更多的投影片投版的例子，請進入 $\text{cW}\LaTeX$ 網頁，退回網頁的上一層，其中有幾個教學網頁，如總體經濟學，其內有許多 `beamer` 投影片之檔案可供參考。

13.2 信函

除了 `article` 與 `book` 兩個文稿類別之外，上一節介紹的是 `beamer` 文稿類別。 \LaTeX 還有一套可排版信函之 `letter` 文稿類別。如果常有機會寫信，我們可以在信函上設計一個信頭標識 (letter head)。以下首先介紹 `letter` 文稿類別之指令，再說明如何設計標識信頭。



例 13.1: 信函

13.2.1 letter 文稿類別

使用 letter 文稿類別排版信函, 指令如下:

```

\documentclass[12pt]{letter}
\begin{document}
\begin{letter}{...}
...
\end{letter}
\end{document}

```

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, 信函內會自動加上當天的日期, 其形式如 “September 19, 2005”。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。

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。例 13.1 為排版指令與結果, 檔名為 letter1.ctx, 置於 cwtex 之例子檔案夾內, c:\texmf\cwtex\examples\letter, 可自行排版測試。

例 13.1 左邊是標準之英文書信格式；右上角以 `\address` 排版發信人之姓名與地址，左上方為收信人之姓名與地址，這是信函正文的一部分。信函正文須輸入於 `letter` 指令環境內，並以 `\begin{letter}{...}` 起頭，大括號內即輸入受信人之姓名與地址。如果不輸入姓名與地址，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。

一個信函檔案內可以同時排版數封信函，但每封信函都須置於 `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。若排版當天為 2005.9.20，則日期格式將為“2005/9/20”之格式。若把指令中的右斜線 / 改成英文句點，排版結果將變成“2005.9.20”。

信函正文可以使用一般 \TeX 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 \TeX 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較方便。以上之英文標題都可以重新定義，請見 14.4 節（頁 252）之說明。

13.2.2 設計信頭標識

如果常有信件來往，我們可以設計一信頭標識置於信函第一頁上端。`letter` 文稿類別只要略加修改，就可以加入信頭標識，最簡單的方法如下。先以 \TeX 排版出信頭標識之設計，將之轉換成 EPS 圖形之後，再引入 `letter` 文稿類別內。

圖 13.2 是設計信頭標識的一個例子。此一設計以一橫線將信頭區分兩部分，上方左邊為機構之商標 (logo) 與中英文名稱，右邊為中文名字與地址。橫線下方左邊為英文名字與通訊地址，右邊則為 email 及電話。由 \TeX 排版指令的概念來看，此一信頭標識除了橫線之外，其餘部分是由 4 個方塊組成；每一個方塊都是以 `\parbox` 指令排版。以左上角之方塊為例，第 10 行之指令排出三條斜線之

Tsong-Min Wu, Professor
Department of Economics
21 Hsueh Road, Taipei, Taiwan

E-mail: example@ntu.edu.tw
Fax: (02)2321-5704
Tel: (02)2351-9641 x 523

```

1 \documentclass[a4paper]{article}
2 \usepackage{mathpazo}
3 \usepackage{graphicx,color}
4 \thispagestyle{empty}
5 \definecolor{dark}{gray}{0.4}
6
7 \begin{document}
8 \begin{minipage}{\textwidth}
9 \parbox[b]{.5\textwidth}{
10 \textcolor{dark}{\huge \textsf{\textit{III}}}} ~
11 \textcolor{dark}{\huge\ctxfrb 台大經濟系}\[2pt]
12 \textsc{\large National Taiwan University}\hfill%
13 \parbox[b]{.45\textwidth}{\raggedleft\small
14 吳聰敏\[-1pt]
15 台大經濟系教授\[-1pt]
16 台北市徐州路21號}\[-6pt]
17 \hspace*{-.04\textwidth}\rule{1.08\textwidth}{.4pt}\[-2pt]
18 \parbox[t]{.5\textwidth}{
19 \textbf{Tsong-Min Wu}, Professor\[-3pt]
20 Department of Economics\[-3pt]
21 21 Hsueh Road, Taipei, Taiwan}\hfill%
22 \parbox[t]{.4\textwidth}{\raggedleft
23 E-mail: example@ntu.edu.tw\[-2.5pt]
24 Fax: (02)2321-5704\[-2.5pt]
25 Tel: (02)2351-9641 x 523}
26 \end{minipage}
27 \end{document}

```

圖 13.2: 設計信頭標識: letterhead.ctx

商標圖，這是以 sans serif 字體之斜體字型排版 3 個大寫字母 I 而來。

第 11 行指令接著以粗圓體排版商標右方之「台大經濟系」5 個字。因為中文粗圓體之筆畫較粗，故其筆畫是以 0.4 之灰階色排版。機構之中文名稱之下再排版英文名稱 National Taiwan University。本例第 2 行引入 mathpazo 數學字體巨集套件，其英文字體是有名的 Palatino，字體相當優美。

橫線上方右邊之方塊也是以類似之指令排出，但以 \raggedleft 指令（第 13 行）讓 3 行文字靠右編排。橫線上方左右兩個方塊之間加入 \hfill 指令（第 12 行），以產生分居兩邊之效果。橫線下方之兩個方塊也是以類似的指令排版，橫線本身則以 \rule 指令排版（第 17 行）。因為我們設計讓橫線凸出文字版面之左右沿，故 \rule 指令之前利用 \hspace* 指令，將線條往左拉出文字版面寬度的

4%。圖 13.2 之例子存檔為 `letterhead.ctx`。排版之後，可產生一 EPS 檔案，檔名為 `letterhead.eps`。

接下來，我們說明如何將 `letterhead.eps` 納入排版信函之文稿類別內。以 `letter.cls` 檔案為基礎，我們很容易可以建立另一個文稿類別檔案。cwTeX 提供一個例子，檔名為 `cwletter.cls`，儲存於 `c:\texmf\cwtex\examples\letter` 檔案夾內，內容如圖 13.3 所示。引用信頭標識之指令是在檔案的第 13 行。設計信頭標識時，我們要求橫線凸出版面左右沿各 4%，故引入信頭標識，其長度定為 `\textwidth` 的 1.08 倍。例子檔案夾內另有 `letter2.ctx` 檔案，其內引用了 `cwletter.cls` 文稿類別。若排版此一信函，第一頁的上方即排出圖 13.2 之標識。

如果你有意參考以上例子設計信頭標識，可由例子檔案夾內找出信頭標識之檔案 `letterhead.ctx`。先複製一份，另取一新檔名，如 `mylh.ctx`。開啓檔案並作必要的修改，排版之後在 `c:\xtemp` 之下將產生 `mylh.ps`。欲將此標識以外製圖形方式引入信函內，我們須將之轉換為 EPS。請先進入 DOS，再進入 `c:\xtemp`，執行以下指令即轉換出 `mylh.eps` 及 `mylh.pdf`：

```
c:\xtemp>makeeps mylh
```

請確你的電腦已安裝 perl，否則 `makeeps` 無法運作。將這兩個圖形檔複製回原檔案夾，或者任何 MiKTeX 可搜尋到之檔案夾內。

設計好 `mylh.eps` 之後，可依以下步驟製作個人信函之文稿類別。

- 將 `cwletter.cls` 複製為 `myletter.cls` (或任何名字)，
- 以文字編輯軟體開啓，第 2 行之 `cwletter` 改為 `myletter`，
- 若信頭標識檔名為 `mylh.eps`，第 13 行之 `letterhead.eps` 應改為 `mylh.eps`。

存檔後，請更新 MiKTeX 之檔案資料：

```
開始 > MiKTeX > MiKTeX Options
```

按“Refresh Now”方塊即完成更新。

13.3 大宗信函

欲編排較複雜的大宗信函 (bulk mails)，可使用 Mike Piff 所寫的 `textmerg`。排版大宗信函必須準備兩份檔案，一為信函內容，一為地址檔案。信函正文之格式類

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesClass{cwletter}
3 \RequirePackage{ifthen}
4 \RequirePackage{graphicx,color}
5 \voffset=-2.5cm
6
7 \DeclareOption*{\passOptionsToClass{\CurrentOption}{a4paper}}
8 \ProcessOptions
9 \LoadClass[12pt,a4paper]{letter}
10 \renewcommand{\today}{\number\month-\number\day-\number\year}
11
12 \newsavebox{\ltrhead}
13 \sbox{\ltrhead}{\includegraphics[width=1.08\textwidth]{letterhead.eps}}
14
15 \renewcommand{\ps@firstpage}
16   {\setlength{\headheight}{3cm}\setlength{\headsep}{.4cm}%
17    \renewcommand{\@oddhead}{\hspace*{-10mm}%
18     \raisebox{-1mm}{\usebox{\ltrhead}}}}
19
20 \renewcommand{\ps@headings}
21   {\renewcommand{\@oddhead}{}}
22 \renewcommand{\@oddfoot}{}
23 \renewcommand{\@evenhead}{\@oddhead}
24 \renewcommand{\@evenfoot}{\@oddfoot}
25 \providecommand{\@evenhead}{}\providecommand{\@oddhead}{}
26 \providecommand{\@evenfoot}{}\providecommand{\@oddfoot}{}
27
28 \pagestyle{headings}

```

圖 13.3: 個人信函文稿類別: cwletter.cls

似一般的 letter 文稿類別, 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 13.4 的左邊為信函內容 (bulkmail.ctx), 右邊為地址檔 (address.ctx), 這兩個檔案都儲存於 c:\texmf\cwtex\examples\letter 檔案夾內。以文字編輯軟體開啓 bulkmail.ctx, 直接排版之後可得 3 頁之信函, 各頁信函之收信人姓名與地址分別取自 address.ctx。底下簡單說明 textmerg 之運作原理。

排版信函時, \TeX 須自地址檔案中取用 4 行之資訊: (1) 收信人頭銜, (2) 地址, (3) 姓氏, 及 (4) 空白行。我們特別定義一空白行, 目的是讓地址檔案內所排列之資訊易於區分。信函檔案之第 7 行為:

```
\Fields{\Title\Add\Surname\en}
```

括號內之 \Title 即對應 (1) 收信人頭銜, \Add 對應 (2) 地址, \Surname 則對應 (3) 姓氏。最後, \en 對應空白行。經過以上之定義, 信函內即以 \Title 排版收

1	<code>\documentclass{cwletter}</code>	主任
2	<code>\usepackage{textmerg}</code>	台中商專\\ 台中市三民路
3	<code>\signature{吳聰敏}</code>	簡
4	<code>\includeonly{}</code>	
5	<code>\begin{document}</code>	主任
6	<code>\include{address}</code>	糖業研究所\\ 台南市生產路
7	<code>\Fields{\Title\Add\Surname\en}</code>	蕭
8	<code>\Merge{address.tex}{%</code>	
9	<code>\begin{letter}{\Surname\Title\\ \Add}</code>	主任
10	<code>\opening{\Surname 主任: ~ 您好!}</code>	高雄工專\\ 高雄市建工路
11	<code>\fontsize{12}{22pt}\selectfont</code>	趙
12	陳中文先生擬申請貴校教職,	
13	本人很高興推薦之...	
14	敬請\Surname 主任惠予考慮。	
15	肅此, ~ 順祝	
16	<code>\closing{研安!}</code>	
17	<code>\end{letter}}</code>	
18	<code>\end{document}</code>	

圖 13.4: 大宗郵件: textmerg 巨集套件

信人頭銜。以上之指令名稱可任取。本例之地址名稱並不長, 故在 `address.ctx` 檔案內是輸入為一行。若地址較長, 我們可以將之拆為兩部分, 譬如, 門牌與街名可能以 `\street` 代表, 城市則以 `\city` 表示。

信函檔案第 1 行使用 `cwletter` 文稿類別排版, 第 2 行引入 `textmerg` 巨集套件。信函之內容須置於 `\Merge` 指令內 (第 8–17 行)。`\Merge` 指令首先引入地址檔案 (`address.tex`), 之後則以 `letter` 指令環境所排版之信函內容, 信函內容會出現在每一封信件內。

為方便處理地址檔案, 檔案之第 4 行加入 `\includeonly{}` 指令, 第 6 行則加入 `\include{address}` 指令。排版信函時, `cwtex` 程式只轉換了信函本身之中文, 但地址檔案內還有一些中文字。若後者之中文不作轉換, `latex` 程式無法排版。以上兩道指令中, `\include{address}` 指令讓 `cwtex` 轉換地址內之中文, `\includeonly{}` 指令則指示不須把轉換結果引入信函內。

若地址檔名為 `myaddress.ctx`, 則第 6 行指令應改為 `\include{myaddress}`。請注意, 地址檔案之附加檔名必須是 `.ctx`, 而且, `bulkmail.ctx` 與 `address.ctx` 須置於同一檔案夾內。

排版時, 以文字編輯軟體開啓 `bulkmail.ctx`, 點選 `cwTeX` 圖像即執動 `cwtex` 程式。因為 `\include{address}` 指令之作用, `cwtex` 程式同時也將地址檔案內之中文轉換為 `TeX` 字型碼, 並另存為 `address.tex`。第 8 行 `\Merge` 指令即指示讀取此地址檔案。另外, 第 4 行 `\includeonly{}` 指令之作用是防止 `TeX` 把地址檔

之內容也編排於信函內。

13.4 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、唱片或錄音帶之標籤等，都可以用此巨集套件編排。

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。台灣同時通行 letter size 與 A4 兩種規格之紙張尺寸。如果你使用的是 letter size 紙張，請在文稿類別指令中加上 `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 13.5 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 13.5 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 \LaTeX 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.ctx` 檔案內，排版之指令如下：

```
\usepackage{labels}
\includeonly{}
\begin{document}
\include{names}
```

```

\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
台南市生產路 54 號
\end{labels}
\end{document}

```

圖 13.5: 固定格式標籤

```
\labelfile{names.tex}
```

請注意, `\labelfile` 指令並不須置於 `labels` 指令環境內。以上之檔案取名為 `label2.ctx`, 存於例子檔案夾內, 請自行測試。

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自動轉換為 \TeX 指令。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, 並儲存為 `names.tex`。第 6 行指令即引入此轉換後之檔案。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 13.6 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
```

```

\documentclass[11pt,letterpaper]{article}
\usepackage{labels}
\LabelRows=10
\numberoflabels=30
\voffset=1.5cm
\LeftBorder=1.2cm
\TopBorder=1.3cm
\BottomBorder=1.2cm
\begin{document}
\fontsize{11}{12pt plus.3pt minus.2pt}\selectfont
\addresslabel{台大經濟系\
台北市徐州路 21 號\
(0)02-2351-5468\
(F)02-2351-1826}
\end{document}

```

圖 13.6: 重覆內容之標籤

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。

13.5 考題與解答

學校的教師經常出習題與考題給學生, 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, 習題或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 13.7 是一份考題的簡化版本, 原檔案為 `answer1.ctx`, 置於 `cwTeX` 例子檔案夾內。第 1 行引入 `answers` 巨集套件; 第 2 行指令如下:

```
\Newassociation{sol}{Solution}{ans}
```

第一選項創造一指令環境 `sol`, 文稿內之考題解答即輸入於此指令環境內。本例子以 `enumerate` 指令環境排版出一大題的兩小題。每一小題以 `\item` 指令輸入, 其下即以 `sol` 指令環境輸入解答。以上指令中之 `Solution` 也是一個新創造之指令環境, 其目的是用於設計習題之排版格式。如果接受內定之格式, 我們不須引用此一指令環境。

```

1 \usepackage{answers}
2 \Newassociation{sol}{Solution}{ans}
3 \Opensolutionfile{ans}[ans1]
4
5 \begin{document}
6 {\large\ctxxf 經濟學原理期末考}
7
8 \begin{enumerate}
9 \item 某國之 M1B 供給額原為1,200億元，其中通貨淨額為200億元，...
10 \begin{enumerate}
11 \item 請問準備貨幣之變動為何？
12 \begin{sol}
13 準備貨幣不變。
14 \end{sol}
15
16 \item 假設利率下降之後，通貨淨額/活存比率 ( $\$C/D\$$ ) 變為0.3，
17 請計算 M1B 等於多少？
18 \begin{sol}
19  $[\mbox{M1B} = \mbox{mm}] \times \mbox{MB}_0 \ ]$ 
20  $\$ \mbox{MB} = 200+100=300\$$ 。
21 \end{sol}
22 \end{enumerate}
23
24 \item ...
25 \end{enumerate}
26 ...
27 \newpage
28 \Closesolutionfile{ans}
29 \input{ans1}
30 \end{document}

```

圖 13.7: 考題與解答之一

指令 `\Newassociation` 的第3選項為 `ans`。排版時，`answers` 巨集套件會在硬碟內創造出一個檔案，檔名為 `ans.tex`，其內即儲存習題解答內容。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`，則文稿內輸入解答時，須置於 `answer` 指令環境內。

以上之指令選定解答之檔案名稱為 `ans.tex`，實際排版時須下一指令要求電腦開啓此檔案，以便接受輸入之解答。圖 13.7 第3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，若排版教科書，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，則開啓兩個

不同名稱的檔案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 20 章，每一章之解答可以存入各自之檔案內以利分別處理。

解答輸入完成之後，須作結束 (close) 檔案之動作。故圖 13.7 之第 28 行加入以下指令：

```
\Closesolutionfile{ans}
```

以結束檔案。接下來，我們以 `\input{ans1}` 指令將前面已儲存之習題解答引入文稿內。排版時，解答將依原先題目之格式出現。本例是以 `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，我們可以重新設定。譬如，在全文設定區加入下列一行指令：

```
\renewcommand{\solutionstyle}[1]{\fbox{#1.}}
```

解答編號將以標準字體排版，並加上一四方形，如 **1**。巨集套件 `answers` 的原始設定是 `\textbf{#1}`，若不加設定，編號數字即以粗體字排版。

圖 13.8 (頁 244) 是另一個例子，檔名為 `answer2.ctx`。此一例子之內容與上一個例子相同，但我們把單數題與雙數題分開儲存。首先，指令第 2–3 行定義兩個指令環境，取名為 `so` 與 `se`。第 14–16 行以 `so` 指令環境輸入第 1 個解答，20–23 行則以 `se` 指令環境輸入第 2 個解答。指令第 4–5 行開啓兩個檔案以儲存解答，分別取名為 `so1.tex` 與 `se1.tex`。最後，第 32 與 34 行指令則將解答排版於分開的兩頁內。

```

1 \usepackage{answers}
2 \Newassociation{so}{Solution}{anso}
3 \Newassociation{se}{Solution}{anse}
4 \Opensolutionfile{anso}[so1]
5 \Opensolutionfile{anse}[se1]
6
7 \begin{document}
8 {\large\ctxff 經濟學原理期末考}
9
10 \begin{enumerate}
11 \item 某國之 M1B 供給額原為1,200億元，其中通貨淨額為200億元，...
12 \begin{enumerate}
13 \item 請問準備貨幣之變動為何？
14 \begin{so}
15 準備貨幣不變。
16 \end{so}
17
18 \item 假設利率下降之後，通貨淨額/活存比率 ( $\$/D\%$ ) 變為0.3，
19 請計算 M1B 等於多少？
20 \begin{se}
21 \[ \mbox{M1B} = \mbox{mm}\times \mbox{MB。} \]
22  $\$ \mbox{MB} = 200+100=300\%$ 。
23 \end{se}
24 \end{enumerate}
25
26 \item ...
27 \end{enumerate}
28 ...
29 \newpage
30 \Closesolutionfile{anso}
31 \Closesolutionfile{anse}
32 \input{so1}
33 \newpage
34 \input{se1}
35 \end{document}

```

圖 13.8: 考題與解答之二

14 巨集指令

專業排版必須照顧到許多細節。T_EX 提供豐富的排版指令，讓我們能控制版面的所有細節。不過，從排版者的角度來看，我們希望能以一兩個指令就排出適當的章節標題，註解與表格等，而不要自己輸入一連串複雜的指令。巨集指令 (macros) 的概念乃因應而生。

巨集指令是把一連串指令集合在一起，並給予一個名稱。文稿內若使用巨集指令，排版時 E_TX 將依序執行其中的指令。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級與字體。欲達成此一目標，最有效率的方法是使用巨集指令。

E_TX 本身事實上就是由 T_EX 原始指令所定義出來的一套龐大的巨集指令。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。

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}
```

E_TX 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 E_TX 本身是由 T_EX 指令組合而成，因此，我們也可以使用 T_EX 指令定義巨集指令。若使用 T_EX 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 `\leref` 設定一個排版參考文獻的格式：

```
\newcommand{\leref}{\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，`\leref` 應輸入於每一項文獻前端，文獻的第一行將由版面左沿開始編排，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` 標識符號是由下列巨集指令所定義：

```
\newcommand{\cw}{\texttt{cw}\TeX}
```

定義一開始首先選用 `typewriter` 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`。為何鍵入 `\cw` 指令時，其後要加上左右大括號？請見 5.1 節（頁 55）之說明。

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， \TeX 排版摘要時，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}
```

巨集指令可加入變數。譬如，使用 \TeX 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,其中的 [1] 設定表示使用者須自行加入一數字。排版時,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。

巨集指令可加入之變數不限於一項。若有兩項變數,則巨集指令名稱之後須改成 [2],定義內則分別以 #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 139)。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\! \vphantom{\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 [Mittelbach and Goossens \(2004\)](#)。

14.1.1 設定字級之巨集指令

文稿內經常須變更字級與行距。 \TeX 提供相對字級指令,如 `\small`, `\large` 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,我們可以將以上指令定義於一巨集指令內,請見圖 14.1。

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。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}

```

圖 14.1: 設定字級與行距之巨集指令

改變字級時，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。若覺得以上的行距不妥，可以改用其他數字。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 之字體只須鍵入 \sz14 指令即可。選用 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。

巨集指令須輸入於文稿的全文設定區。但是，若每一篇文稿都要輸入巨集指令定義，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，儲存於 $\text{cwt}_\text{E}\text{X}$ 之例子檔案夾內。因此，我們只須在全文設定區輸入：

```
\input{mymacro}
```

文稿內即可使用 \sz12 等指令改變字級與行距。

排版時， $\text{E}\text{T}_\text{E}\text{X}$ 從頭讀取文稿，一旦遇到 \input{...} 指令，馬上讀入該檔案，立即處理其內容。因此，\input 指令之效果等於是將 mymacro.tex 之內容全部輸入於下指令之位置。換言之，\input 指令的目的也是在簡化輸入：把一大串指令另置一處，以供隨時取用。

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，但使用上須注意以下三點：

- 含中文之巨集檔案須以 .ctx 為附加檔名;
- 以 \input 或 \include 指令引入巨集檔案時, 請勿加入附加檔名。
- 巨集檔案須與主檔案置於同一檔案夾內, 或置於工作子目錄 c:\xtemp 內。



中文巨集檔案

例如, 若主檔案名為 paper.ctx, 其中引用含有中文之巨集檔案 chmacro.ctx, 則後者須置於 paper.ctx 之檔案夾內, 或者 c:\xtemp 內。另外請注意, 以 \input 與 \include 引入檔案時請勿加入附加檔名。cwtex 轉換中文時, 若發現文稿內有 \include{chmacro} 或 \input{chmacro} 指令, 它會在目前檔案夾 (current directory) 內尋找 chmacro.ctx 檔案。若檔案存在, 即將之轉換為 chmacro.tex。接下來 latex 進行排版工作時, 即讀取 chmacro.tex。

巨集檔案內若僅含英文, 附加檔名可取為 .tex, 並且可置放於 \TeX 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 為附加檔名, 並且要與文稿檔案置於同一檔案夾或 c:\xtemp 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文字版面寬度, 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, 讓圖表右沿對齊文字版面右沿。若圖表是出現在單數頁 (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, 我們須判斷本頁是單數頁還是雙數頁, 並依此作進一步的處理。

前面 7.7.2 節 (頁 104) 曾介紹 \TeX 之 \ifodd 與 \count0 指令, 可判斷頁碼是奇數還是偶數。以下將所介紹的 ifthen 巨集套件 (作者 David Carlisle), 也可以處理類似的問題。舉例來說, cw \TeX 所提供之書籍排版例子使用了 cx-macro.ctx 巨集指令, 其內定義一 \btable 巨集指令如下:

```
\usepackage{ifthen}
...
\newcommand{\btable}[2]{
\begin{table}\label{#1}
\ifthenelse{\isodd{\pageref{#1}}}{\noindent\ignorespaces}%
{\noindent\hspace*{-.2\textwidth}\ignorespaces}
\fbboxsep=.05\textwidth
\colorbox{hlight}{\ignorespaces}
\begin{minipage}{1.1\textwidth}
\centering
#2
\end{minipage}}
\end{table}}
```

此巨集指令使用了 `ifthen` 巨集套件所提供之 `\ifthenelse` 指令，並且用了兩個參數。指令一開始以 `\label` 指令下一標籤，並以 `\pageref` 讀出頁碼（指令第 3 行）。有了頁碼之後，再以 `\isodd` 指令判斷是否為奇數頁。

若為奇數頁，排版動作為指令第 3 行末端之 `\noindent\ignorespaces`，亦即，行首不內縮，並且忽略輸入的文字或指令中的任何空格。反之，若為偶數頁，則以 `\hspace*` 指令將文字圖表左移選定之距離。使用巨集指令時，須填入兩個參數，第一個參數是標籤，第 2 個參數為排版表格之指令。因此，文稿內若輸入以下指令（請見例子檔案夾內之 `cx-gdp.ctx`）：

```
\btable{gdp-ex}{
\caption{衡量國內生產毛額}
\begin{tabularx}{\textwidth}{@{}Xrrrrrrr@{}}
...
\end{tabularx}
```

排版時， \TeX 根據標籤 `gdp-ex` 判斷是在單數頁或雙數頁。若在單數頁，則取消行首內縮，故表格往右凸出一點。反之，若位於雙數頁，則左移文字版面寬度的 20%，再排出表格，故表格右沿將與文字版面右沿切齊。

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 `num`，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{num}{\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{mym}{[2][10mm]{\par\noindent\hspace*{#1}%
\begin{minipage}{#2\textwidth}}{\end{minipage}}
```

此一指令環境的目的是設定迷你版面的大小，並調整其位置。首先，指令環境名稱為 `mym`，其後之 `[2]` 表示此指令環境帶有二項參數。但是，第一項參數是選擇

表 14.1: 計數器

part	paragraph	figure	enumi
chapter	subparagraph	table	enumii
section	page	footnote	enumiii
subsection	equation	mpfootnote	enumvi
subsubsection			

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{mym}{0.8} ... \end{mym}
```

則鍵入之文字內容將排版於迷你版面內, 其寬度為正文行長的 80%。排版時, \LaTeX 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{mym}[4mm]{0.8} ... \end{mym}
```

則迷你版面離正文版面左沿僅 4mm, 而非內定值之 10mm。

14.3 計數器

\LaTeX 定義許多計數器 (counter), 用以記錄章節、註解、方程式之編號。全部的計數器名稱如表 14.1 所示。大部分之計數器名字一看就了解其意義, 唯一須解釋的是最右欄的 4 個計數器。其中, `enumi` 計數器記錄條列指令環境第一層項目之編號; `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編號。

欲重新設定計數器之值, 可使用 `\setcounter` 指令, 譬如:

```
\setcounter{footnote}{20}
```

可將註解計數器定為 20。當碰到下一個 `\footnote` 指令時, 計數器將增加 1, 編號變成 21。書籍若是分章編排, 也可以透過 `chapter` 計數器設定每章編號。例如, 某章為第 8 章, 則該章之前端可加上: `\setcounter{chapter}{7}`。碰到下一個 `\chapter` 指令時, 計數器加 1, 變成 8。我們也可使用 `\addtocounter` 指令將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:

表 14.2: 特定標題之指令與內定值

指令	英文標題	指令	英文標題
<code>\abstractname</code>	Abstract	<code>\indexname</code>	Index
<code>\appendixname</code>	Appendix	<code>\listfigurename</code>	List of Figure
<code>\bibname</code>	Bibliography	<code>\listtablename</code>	List of Table
<code>\ccname</code>	cc	<code>\partname</code>	Part
<code>\chaptername</code>	Chapter	<code>\prefacename</code>	Preface
<code>\contentsname</code>	Contents	<code>\refname</code>	References
<code>\enclname</code>	encl	<code>\seename</code>	see
<code>\figurename</code>	Figure	<code>\alsiname</code>	see also
<code>\headpagename</code>	Page	<code>\tablename</code>	Table
<code>\headtoname</code>	To (letter)		

```
\addtocounter{footnote}{-2}
```

某些計數器名字前加上 `\the` 即變成指令。例如, `\thepage` 即為排版頁碼之指令, 大體而言, 此指令意義與 `\arabic{page}` 相同。同理, `\thesection` 為排版節編號之指令, 而 `\thetable` 可排版表編號。不過, 以 book 文稿類別排版時, 圖表編號之前會自動附加章編號。譬如, 「表 14.1」的第一個數字為章編號, 第 2 個數字才是表編號。這表示 `\thetable` 之定義內不單純只是 `\arabic{table}` 而已。如果不要附加章編號, 只須在全文設定區加入以下指令即可:

```
\renewcommand{\thetable}{\arabic{table}}
```

14.4 中文標題

\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。 \LaTeX 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, \LaTeX 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}{\ctxfr 表}
```

英文 Table 即改為「表」, 並以圓體字排版。

表 14.2 列出所有的標題指令, 及內定之英文標題, 其中, `\enclname`, `\ccname`, `\headtoname` 與 `\headpagename` 4 個指令是用於 letter 文稿類別; `\seename` 與 `\alsiname` 則用於索引。

15 參考文獻

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。專業排版的基本要求是正確，但徵引文獻的過程卻相當繁瑣，容易出錯。譬如，文稿內原本徵引某篇文獻，後來改變心意，但卻忘了自文末之參考文獻中刪除。反之，修改文章時決定加入某一文獻，但卻忘了加入文末之參考文獻內。

為解決排版參考文獻的問題，Oren Patashnik 發展出一套系統，稱為 BiB_T_EX，其中包含了 `bibtex` 程式及一些工具檔案。不過，BiB_T_EX 僅能處理英文或其他西洋文字。為了排版中文參考文獻，我們須延伸 `bibtex` 程式之功能，同時還須設計排版中文參考文獻之格式。cw_T_EX 所提供之程式稱為 `cwbibtex`，它能同時處理中英文參考文獻。

15.1 排版參考文獻

在介紹參考文獻之排版流程之前，我們先說明 BiB_T_EX 之原理。首先，排版者須把參考文獻鍵入於文獻資料庫檔案，此檔案須以 `.bib` 為附加檔名。第 2 章例 2.5 (頁 32) 是一個排版參考文獻的例子，第 57 行所列的 `cx3test` 即為文獻資料庫檔案。資料庫檔案內列出個人所用之參考文獻，每一文獻須取一檢索名 (key)。文獻資料庫檔案內容之格式，詳見 15.2 節。

例 2.5 之第 16 行以 `\citet{Wan:2000}` 指令徵引文獻，其中 `Wan:2000` 即為檢索名。同理，第 17 行以 `\citet{Hamilton:89}` 指令徵引另一篇文獻。排版參考文獻之流程如下：

- 建立一參考文獻資料檔，輸入欲徵引之文獻，每一文獻取一檢索名，
- 文稿內以檢索名稱徵引文獻，如 `\citet{Wan:2000}`，
- 排版時，`cwbibtex` (或 `bibtex`) 由文獻資料檔內擷取所引用之文獻，排版於文稿末端；文稿內並自動加入徵引文獻之文句。

由以上之說明可知，參考文獻之排版牽涉到兩種格式，一個是論文末端參考文獻之排列格式，另一個是論文內引用文獻之文句格式。前一格式是由文獻資料格式 (bibliography style) 控制，其附加檔名為 .bst。CTAN 上可找到各式各樣的格式檔案，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。第 2 章例 2.5 所使用的 ter.bst 格式可處理中英文之文獻，這是台大《經濟論文叢刊》所使用之格式。

綜合以上所述，若文獻資料庫檔名為 cx3test.bib，文獻資料之排版格式為 ter.bst，而文稿內徵引文獻時擬使用 natbib 巨集套件之指令，則排版時所需之相關指令如下：

```
\documentclass[12pt,a4paper]{article}
\usepackage{natbib}
\renewcommand{\refname}{\ctxfbb 參考文獻}
...
\begin{document}
...
\bibliographystyle{ter}
\bibliography{cx3test}
\end{document}
```

以上第 2 行指令引用 natbib 巨集套件，15.3 節 (頁 262) 將介紹使用方法。文稿以 article 文稿類別排版時，文末排版參考文獻之前會自動加入英文標題 **References**。第 3 行指令將之改為「參考文獻」四字，並選用粗黑字體。若文稿是以 book 文稿類別排版，英文標題為 **Bibliography**，更改指令為：

```
\renewcommand{\bibname}{\ctxfbb 參考文獻}
```

本例中，倒數第 2 行指令選用文獻資料檔 cx3test.bib。若你個人之文獻資料檔取名為 mybib.bib，應改為：

```
\bibliography{mybib}
```

請注意，不要輸入附加檔名。

若上例之文稿檔名為 paper.ctx，排版方式如下。依序執行 cwtex 及 latex 程式之後，若一切無誤，接下來須執行 cwibibtex 程式。若是使用 WinEdt 排版，在上方工作列上 GSview 圖像的右邊是 YAP 圖像 (其上是一放大鏡加 DVI 英文字)；

再右邊就是 `cwibibtex` 之圖像，其上有 Bib 英文字。點選 `cwibibtex` 圖像，電腦即執行 `cwibibtex` 處理參考文獻。(如果你使用 `Crimson editor`，請按 `Ctrl-[F12]` 啟動 `cwibibtex`。) 若無錯誤，接下來以 `dvips` 轉換為 PS 檔案，最後再以 `GSview` 預覽。`CTEX` 之例子檔案夾 `c:\texmf\cwtex\example` 內有 `examp5.ctx`，其內含排版參考文獻之指令，可供測試。

15.1.1 設定選項

以上點選 `cwibibtex` 圖像時，啟動的是一批次檔案，`cwibibtex.bat`，內容如下：

```
@echo off
rem New version of cwibibtex (2005.8)
copy %1.ctx c:\xtemp
c:
cd \xtemp
cwibibtex.exe -s2 -n %1.ctx
latex %1
latex %1
```

處理參考文獻時，`cwibibtex` 須讀取文稿原檔。若檔名為 `paper.ctx`，第 3 行指令將 `paper.ctx` 檔案複製至工作檔案夾 `c:\xtemp` 內以方便後續處理。第 4-5 行則進入 `c:\xtemp` 檔案夾內，第 6 行是啟動 `cwibibtex` 程式。

執行 `cwibibtex` 程式時可加入選項。首先，參考文獻通常是排列於文稿末端。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，則第 6 行指令內請在檔名 `%1.ctx` 之前加入 `-e` 選項。其次，排版於文稿末端之文獻都已經過排序。目前，`cwibibtex` 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 Big5 之順序。如果要使用第 2 種排序法，請加入選項 `-s2`，這是批次檔之內定值。

除此之外，參考文獻是以文稿所設定之行距排版。但若是中英文參考文獻並存，英文文獻之行距應較中文文獻為小。如本例所示，若加入 `-n` 選項，英文文獻將以 `TEX` 內定之英文行距排版，並且中英文文獻之間會空出一點距離。本例中，選項 `-s2` 置於 `-n` 之前，但實際上，順序並不重要。

`cwibibtex` 程式之功能是從文獻資料檔案擷取所需之文獻，並作分類與排序。如果執行過程沒有錯誤，接下來，我們須執行 `latex` 兩次，才能排版出參考文獻。因此，`cwibibtex` 批次檔的最後兩行是執行 `latex` 程式。如果你習慣使用 `pdflatex` 排版，最後兩行之 `latex` 應改為 `pdflatex`。

15.1.2 搜尋文獻資料檔

排版時, `cwibibtex` 程式會到特定檔案夾內尋找 `.bib` 檔案。首先, 它會搜尋工作子目錄 `c:\xtemp`。若工作子目錄內找不到 `.bib` 檔, 它會到 `c:\texmf\bibtex` 以及 `c:\localtexmf\bibtex` (以及其下之子目錄) 尋找。若找不到, 它會再前往 `d:\tex` (以及其下之子目錄) 尋找。

文獻資料檔之搜尋路徑可自行設定。以文字編輯軟體開啓 `MiKTeX` 之設定檔 `miktex.ini` (存於 `c:\localtexmf\miktex\config` 子目錄內), 其中 `[BibTeX]` 底下之 `Input Dirs=` 即設定搜尋路徑。英文句點代表目前之子目錄, `%R` 代表 `c:\texmf` 及 `c:\localtexmf`, 而 `//` 則代表以下各層之子目錄。如果你要把 `.bib` 檔案放在特別的檔案夾內, 可重新設定。

另外一種設定的方法是在輸入文獻資料庫檔名時, 同時也輸入檔案夾名稱。譬如, 若文獻資料庫檔名為 `cx3.bib`, 儲存於 `d:\tex\bibtex` 檔案夾內, 則文稿內之指令應為:

```
\bibliography{d:/tex/bibtex/cx3}
```

請特別注意, 檔案夾名稱為右斜線, 而非左斜線。

排版參考文獻時, 我們可能使用數個文獻資料檔。如果我們建立兩個文獻資料庫, 一個是 `cx3test.bib`, 另一個名為 `cwtex.bib`, 而某篇文稿若同時引用其中的文獻, 引用指令如下:

```
\bibliography{cx3test,cwtex}
```

請注意, 你不能把同一文獻輸入在這兩個 `.bib` 檔案內, 否則排版時會出現錯誤。

15.2 建立文獻資料檔

建立文獻資料檔是排版參考文獻的起點。文獻資料檔必須正確無誤, 否則 `cwibibtex` 處理時會發生錯誤。最容易出現的錯誤是少了大括號或逗點。 `cwibibtex` 程式執行時, 相關的訊息會出現在 DOS 視窗內, 但訊息畫面通常是一閃即過, 無法直接閱讀。不過, 全部的訊息會儲存於 `.blg` 檔案內, 我們可以文字編輯軟體開啓此一檔案以檢查處理結果, 以下將進一步說明。

`cx3test.bib` 所儲存的文獻資料檔例子 `cx3test.bib` 置於 `c:\texmf\bibtex\bib` 檔案夾內, 圖 15.1 列出 `cx3test.bib` 文獻資料內 5 項文獻, 前三項是第 2 章例 2.5 (頁 32) 所徵引的文獻。依 `BiBTeX` 之規範, 文獻資料可採兩種格式輸入, 第一種

```

@PREAMBLE{ "\providecommand{\cleftquote}{()} # "\providecommand{ ...
...
@ARTICLE{Wan:2000,
  author = {萬哲鈺},
  title = {中央銀行台北外匯市場干預行為分析},
  journal = {台灣經濟學會年會論文集},
  year = {2000},
  pages = {109--125}
}

@ARTICLE{Hamilton:89,
  author = {Hamilton, James D.},
  year = 1989,
  title = {A New Approach to the Economic Analysis of Nonstationary ...}
  journal = {Econometrica},
  volume = 57,
  pages = {357--384}
}

@ARTICLE{Beine:03,
  author = {Beine, Michel and Laurent, Sebastien and Lecourt, Christelle},
  title = {Official Central Bank Interventions and Exchange Rate ...},
  journal = {European Economic Review},
  year = {2003},
  volume = {47},
  number = {5},
  pages = {891--911}
}

@CUNPUBLISHED{WuTM2003c,
  author = {吳聰敏·駱明慶},
  title = {好老師，莫強求},
  url = {http://ccms.ntu.edu.tw/~ntut019/edu/edu.html},
  year = {2003},
}

@CBOOK{ChangYS1993,
  title = {台灣平埔族文獻資料選集 --- 竹塹社},
  publisher = {中央研究院台灣史田野研究室},
  year = {1993},
  author = {張炎憲·王世慶·李季樺},
  address = {台北},
  note = {上下2冊},
}

```

圖 15.1: 文獻資料庫

格式如圖 15.1 所示，每一項資料都以大括號界定範圍。第二種格式則是以英文雙引號界定範圍。

文獻資料庫可直接以文字編輯軟體編輯，但也可以經由工具軟體輸入。工具軟體中，JabRef 相當有名，使用彈性大，而且可處理中文。如果你未來可能使用 JabRef 軟體輸入整理參考文獻資料庫，我們建議圖 15.1 之格式。另外，須注意的

是每一文獻內之項目應只占一行。圖 15.1 例子中第 2 項文獻 Hamilton:89 與第 3 項 Beine:03 之文章題目 title 都很長,但仍然輸入為一行,以免未來以 JabRef 軟體處理時發生錯誤。

中文參考文獻須特別處理中文標點符號。以文字編輯軟體開啓 cx3test.bib, 第 4 行指令 (亦見圖 15.1 第 1 行) 如下:

```
@PREAMBLE{ "\providecommand{\clefquote}{\} ...
```

此行指令的功能即是處理中文標點符號,包括書名與文稿名之引號。如果你自行製作中文 .bib 檔案,並且以 ter.bst 格式排版,文獻資料檔案內必須加入這一行,否則 cwibibtex 程式無法正確處理。比較簡單的建立 .bib 新檔案的方法是把 cx3test.bib 複製為另一檔案,如 mybib.bib,刪除其中用不著之文獻,再加入自己所需之文獻。

15.2.1 cwibibtex 訊息

文獻資料庫必須正確無誤,否則 cwibibtex 程式無法擷取正確的資料。以例子檔案夾內之 exam5.ctx 為例,執行 cwibibtex 程式之後,c:\xtemp 檔案夾內會產生 exam5.blg,其內容如下:

```
This is BibTeX, Version 0.99cThe top-level auxiliary file: ...
The style file: ter.bst
Database file #1: cx3test@.bib
```

附加檔名 .blg 代表 bibtex log file。因為執行結果正確無誤,故其中並無任何錯誤訊息。

依文獻資料檔之規範,每一文獻內除了最後一行之外,各行必須以逗點結束。如果我們輸入圖 15.1 例子之第一項文獻時,將期刊之年期輸入為:

```
year = "2000"
```

亦即,我們疏忽了最後的逗點。排版 exam5.ctx,並執行 cwibibtex 之後,錯誤訊息將記錄於 exam5.blg:

```
This is BibTeX, Version 0.99cThe top-level auxiliary file: tmp@.aux
The style file: ter.bst
Database file #1: cx3test@.bib
I was expecting a ',' or a '}'---line 31 of file cx3test@.bib
:
: pages = "109-125"
```

```
(Error may have been on previous line)
I'm skipping whatever remains of this entry
(There was 1 error message)
```

第4行訊息說明, `cwbibtex` 預期每一行資料應以逗點或右大括號結束, 下一行則指出錯誤之所在。此時, 我們應開啓 `cx3test.bib`, 搜尋 "109-125" 字串, 訂正錯誤之後再重新排版。

15.2.2 文獻資料檔格式

依 $\text{Bi}\text{T}_{\text{E}}\text{X}$ 之設計, 參考文獻種類有13種, 包括論文 **article**, 書籍 **book**, 碩士論文 **mastersthesis**, 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**。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。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** 若未填入資料, 並不會有任何問題。

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` 設計之格式排版每一文獻。爲了能同時處理中英文參考文獻, $\text{cW}\text{T}_{\text{E}}\text{X}$ 特別設計 `ter.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**; 中文論文則使用 **carticle**。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 15.1 之例子。



中文文獻

底下列出文獻種類之名稱及其格位。每一文獻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/cincollection 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 `cx3test.bib` 之內容, 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。

- 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, \LaTeX 之註銷指令 `%` 在文獻資料庫內並無註銷之功能。
- 英文姓名可輸入為 "Myers, Wykeham W.", 或者 "Wykeham W. Myers"。兩位以上之作者應以 "and" 隔開, 例如: "Maddison, Angus and Harry van Ooststroom"。
- 作者名 Robert E. Lucas, Jr. 應輸入為 "Lucas, Jr., Robert E."。
- 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",
```

- `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格隔開時。如果你要求以特定之順序排版姓名, 應將姓名置於左右大括號內。
- 兩位以上之中文作者名字以一小黑點隔開, 如 "謝森中·潘盛藩"。若使用 WinEdt 輸入文稿, 則按下 `Ctrl+Shift+d` 即可鍵入小圓點。參見表 3.2 (頁 45) 之說明。
- 英文書名或文稿題目內若含有專名詞, 如 Taiwan 或 Darwin, 輸入時應加上大括號, 變成 `{Taiwan}` 與 `{Darwin}`。原因是, 有些參考文獻之格式會改變英文字母之大小寫。譬如, 某一格式可能要求論文題目以小寫字母排版, 因此格式檔案會將題目內之大寫英文字母改為小寫。但是, 專有名詞應使用大寫字母, 不能變成小寫, 不過, 格式檔案無法作正確判斷。解決的辦法是輸入時加上大括號, 即可防止變更為小寫字母。
- 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。

在網路年代, 許多文獻是置於網路上。圖 15.1 例子之 WuTM2003c 是一電子檔案, 置於網路上。輸入文獻資料檔時, 網址應放入 `url` 格位內。有些網址甚長, 且夾雜特別的符號, 排版時可能出問題。為了處理網址之排版, 我們建議在全文設定區引用 `url` 巨集套件:

```
\usepackage{url}
\def\UrlFont{\rm}
```

第 2 行指令是設定以標準字體排版網址, 若不設定, 則以 `typewriter` 字體排版。



大小寫

依據吳聰敏 (1997) 之 ...	依據\citet{WuTM1997} 之 ...
依吳聰敏 (1997, 頁 30) ...	依\citet[頁 30]{WuTM1997} ...
依 Davidson (1903) 之 ...	依 \citet{Davidson1903} 之 ...
依據吳聰敏之研究 ...	依據\citeauthor{WuTM1997} 之研究 ...
依《第二次糖業記事》(1903) ...	依 \citet{SugarNote2} ...
... 結論 (吳聰敏, 1997)。	... 結論 \citep{WuTM1997}。
... 結論 (吳聰敏, 1997, 頁 30)。	... 結論 \citep[頁 30]{WuTM1997}。

圖 15.2: natbib 巨集套件徵引文獻指令

15.3 natbib 巨集套件

每一種期刊各有其參考文獻之排版格式，論文內引用文獻之格式也不同。舉例來說， \LaTeX 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，排版後論文內引用文獻之前後文為“根據 [7] 之分析 ...”。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 ...”。若使用 author–date 格式，論文末端排版參考文獻時不加上編號，其排列通常是依作者姓名之字母順序。

文稿末端參考文獻之排版是以 bibliography style 格式控制，例如 `ter.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。此巨集套件主要用於設定 author–date 格式，但也可用於 \LaTeX 內定之格式。以下僅簡單介紹 author–date 格式。

15.3.1 徵引文獻之指令

\LaTeX 原有徵引文獻之指令為 `\cite`。為了處理 author–date 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*，通常加於行文末端作為補充說明。上面所舉的例子，“根據 Lucas (1983) 之分析 ...”，是 *textual citation*。“經濟成長率的高低與儲蓄率並無相關 (Lucas, 1983)”則是 *parenthetical citation* 之例子。

圖 15.2 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格。第 2 行與第 7 行之例子說明，指令

之前可加入選項，以排版頁碼或其他文字。

在 author–date 格式下，每一篇文章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。我們可以使用 **manual** 與 **cmanual** 排版此類之文獻。在 `cx3test.bib` 文獻資料檔內有一個例子，書名是《第二次糖業記事》，發行機構是「臨時臺灣糖務局」，出版日期是 1903 年，但無編者或作者名字。在文獻資料檔內，此一出版品是以 **cmanual** 類別輸入，但我們特別加入 `key` 格位 (field):

```
@CMANUAL{SugarNote2,
  title = {第二次糖業記事},
  address = {台北},
  year = {1903},
  key = {{第二次糖業記事}},
  organization = {臨時臺灣糖務局},
}
```

`key` 格位有兩個作用。第一，文稿末端排版參考文獻時，將以 `key` 內容之筆劃排序。第二，文稿徵引文獻時，原應出現作者名字部分，將以 `key` 之內容排版。

無作者名字之英文出版品也可以利用 **manual** 文獻類別。例如，若書名為 *System of National Accounts*，出版於 1993 年，檢索名設為 SNA1993。以 **manual** 類別輸入於文獻資料庫時，`key` 格位應輸入書名。文稿內引用此一文獻時，若依上述中文參考文獻之例子，以 `\citet{SNA1993}` 指令徵引，排版結果如下：

根據 *System of National Accounts* (1993), ...

書名的確可以正確排出，但是，英文排版之規範是書名應以斜體字排版。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{SNA1993}}` (1993), ...

排版之後，書名將為斜體字。

某些文獻在文稿內直接輸入作者名字或書名可能反而較方便。以上面的例子來說，我們可直接輸入書名，而不須使用 `\cite{SNA1993}` 指令。但此一作法的問題是，此項文獻不會列入文末之參考文獻。為解決此問題，我們可在文稿內加上 `\nocite{SNA1993}` 指令，該文獻即納入文稿末端之參考文獻內。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之內容。

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，文獻之間距比正常行距約加大一倍。欲改變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，請參閱其說明檔。

15.4 bibentry 巨集套件

徵引文獻的標準作法是把參考文獻之全文置於文稿最後，文稿內僅引用作者名字及年期。不過，有時候我們須在文稿內寫出文獻之全文，學校教師授課之課程大綱是一個例子。此時，我們可使用 Daly, Patrick W. 所寫的 bibentry 巨集套件。為了處理中文，我們將此巨集套件略作修正，取名為 cwibentry。

圖 15.3 是一個簡單的例子。首先，全文設定區須引入 cwibentry 套件。一般引用參考文獻的方法是在文稿末端加入 \bibliography{cx3test} 設定指令，但此處須改用 \nobibliography{cx3test}，請見圖 15.3 第 3 行指令。此行指令請加在 \begin{document} 指令之後。文稿內引用之文獻是以 \bibentry 指令引入，本例之 6–8 行即引入 3 項文獻。我們將 \bibentry 指令置於 enumerate 環境內，但也可以置於文稿內任何地方。

最後，參考文獻之排版格式仍然以 \bibliographystyle 指令選用格式檔案。以中文稿而言，我們須使用 TERentry.bst 格式，而不能使用排版一般參考文獻之 ter.bst。這也是 cwTeX 所特別提供的格式。

排版方法與排版一般文獻的方式一樣，先執行 cwTeX 與 latex 之後，須執行 cwibibtex 批次檔。若使用 WinEdt，點選 \TeX 圖像之後，再點選 cwibibtex 圖像即完成排版。

15.5 參考文獻工具軟體

排版參考文獻有許多工具軟體可供使用，以下僅簡單介紹 2 種。

1. 臺北州 (1922), 《農業基本調查叢書》, 臺北州
2. Allee, Mark A. (1994), *Law and Local Society in Late Imperial China*, Stanford: Stanford University Press
3. 陳秋坤 (1994), 《清代臺灣土著地權》, 台北: 中研院近史所

```

\usepackage{cwbibentry}
...
\nobibliography{cx3test}
\small
\begin{enumerate}
\item \bibentry{AGSurveyTP1922}
\item \bibentry{Allee1994}
\item \bibentry{ChenCK1994}
\end{enumerate}

\bibliographystyle{TERentry}

```

圖 15.3: bibentry 例子

15.5.1 Jabref 軟體

Jabref 軟體是管理文獻資料庫之圖形界面軟體, 它可以處理中文, 但使用者須自行定義中文文獻之種類, 如 **article**, **cbook** 等。安裝及使用方法請見:

<http://jabref.sourceforge.net/>

15.5.2 製作文獻格式檔

若排版英文文稿, CTAN 上有各種格式的 .bst 檔案供下載使用。不過, 英文格式不見得適用於中文文獻。為了解決中英文文獻並列的問題, 格式檔案內除了 **book** 文獻種類外, 必須有對應的 **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 **btchak** 說明檔。

你也可以製作全新格式的 .bst 檔案。最方便的作法是利用 **makebst** 巨集套件, 作者為 Patrick W. Daly。cwTeX 所提供的 **ter.bst** 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 PostScript 字體

中國於 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，同樣一個字卻有不同的程式寫法。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 Type 1 字體，一般稱為 Type 1 字體。1990 年代初期，Microsoft 公司發展 Windows 作業系統時，同時又發展一套字型程式規格，稱為 TrueType。因為 Windows 系統之普及，目前中文 Windows 系統上使用之字體，絕大部分都是 TrueType 規格。 \LaTeX 與 $\text{pdf}\text{\LaTeX}$ 也可以使用 TrueType 字型排版。

$\text{T}_{\text{E}}\text{X}$ 一開始使用 *Metafont* 字體，但以目前的發展狀況來看，Type 1 字體的使用彈性最大，字型選擇也最多。本章將簡單說明 \LaTeX 如何使用 Type 1 字體排版，並介紹一些與字體有關的工具程式。

16.1 英文 Type 1 字體

字體是排成品質的關鍵。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 Type 1 字體都是商業軟體，但也有少數可以自

表 16.1: PSNFSS 巨集套件

巨集套件	Roman	sans serif	typewriter	Math
mathpazo	Palatino			Palatino + Pazo
mathptmx	Times			Times + Symbol
helvet		Helvetica		
avant		Avant Garde		
courier		Courier		
chancery	Zapf Chancery			
bookman	Bookman	Avant Garde	Courier	
newcent	New Century Schoolbook	Avant Garde	Courier	
charter	Charter			

網路下載, 免費使用。其中, URW++ 公司即捐贈了 35 種字體。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。取得 Type 1 字型檔之後, 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。最簡單的方法還是使用巨集套件。

16.1.1 PSNFSS 巨集套件

我們可以透過 PSNFSS 巨集套件使用一些免費而且品質甚佳的英數字體。此一巨集套件一開始是由 Sebastian Rahtz 所發展, 目前則由 Walter Schmidt 接手。PSNFSS 事實上提供好幾套使用字體之巨集套件, 表 16.1 左欄為各套件之名稱。

如果在全文設定區引用 mathpazo 巨集套件:

```
\usepackage{mathpazo}
```

正文將以 Palatino 字體排版, 數學式則以 Palatino + Pazo 字體排版, 其餘則仍使用 \TeX 之原始字體。同理, 如果使用 charter 巨集套件, 正文將改用 Charter 字體, 其他則使用 \TeX 之原始字體。依表 16.1, 若正文要使用 Times 字體, 可使用 mathptmx 巨集套件。此巨集套件同時改變數學字體。如果你只要改變正文字體, 數學式希望使用 \TeX 原始之字體, 應使用 times 巨集套件。

使用 PSNFSS 的一個問題是, 在中文稿內, 英數字體顯得太大。因此, 我們須把英數字體調整得小一些。以 Charter 字體為例, 在 `c:\texmf\tex\latex\psnfss` 子目錄內有 `ot1bch.fd` 檔案。以文字編輯軟體開啓, 將全部的 `<-> bch` 字串改為 `<-> [.94] bch`。以上所加入的 `[.94]` 設定, 是將原字體縮小為 94%。若此比率仍嫌太大, 還可以再縮小一些。`ot1bch.fd` 檔案是設定 \TeX 字型檔原始之編碼方

表 16.2: 英數字體

txfonts	風格接近 Times, 但加入新設計之數學字體
pxfonts	風格接近 Palatino, 但加入新設計之數學字體
fourier	以 Utopia 字體為正文, 但加入新設計之數學字體
ccfonts	以 Knuth 之 Concrete 數學字體為基礎

式。PSNFSS 之作者建議改用新編碼 T1 與 TS1 (用於文字符號)。若依其建議, 我們還須修改 `t1bch.fd` 以及 `ts1bch.fd`。OT1 之編碼每一個字型檔只考慮 128 個字元, 而 T1 則考慮 256 個字元。這對於排版歐洲國家之文字特別方便。

如果我們是使用 `mathpazo` 巨集套件, 除了調整正文 (Palatino) 字體之外, 也須調整數學字體之大小。Palatino 字體是以 `ppl` 命名, 因此, 我們須以上列方式修改下列數字字體定義檔之內容: `ot1ppl.fd`, `omlppl.fd`, 與 `omsppl.fd`。如使用 T1 編碼, 則 `t1ppl.fd` 與 `ts1ppl.fd` 也須調整。若要了解以上設定之原理, 請參考 16.4 節。

PSNFSS 巨集套件基本上是以軟體公司所捐贈之 Type 1 字體為基礎。另外一種作法是全新設計字體。例如, Young Ryu 設計了兩組數學字體, 第一套稱為 `txfonts`, 第二套稱為 `pxfonts`。前者的字體設計風格接近 Times 字體, 後者接近名字體設計家 Hermann Zapf 所設計的 Palatino 字體。以 `pxfonts` 為例, 我們僅須在全文設定區加入下列指令即可:

```
\usepackage{pxfonts}
```

不過, 此一字體仍較中文字為大, 故須修改 `.fd` 檔案以調整其大小。`pxfonts` 全部的 `.fd` 檔案都儲存於 `c:\texmf\tex\latex\pxfonts` 子目錄內。

表 16.2 列出幾套可供選用之英數字體, 請注意, 如果使用 `fourier` 巨集套件, 請記得加入 OT1 字型編碼指令:

```
\usepackage{fourier}
\usepackage[OT1]{fontenc}
```

否則中文字出不來。以上英數字體巨集套件之詳細介紹, 請見 [Mittelbach and Goossens \(2004\)](#), 第 7 章。

圖 16.1 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。最底下一列之正文字體使用 Adobe Garamond, 數學符號則使用 Mathtime 字體。這兩套字體都是商業軟體, 後者是由 Y&Y 公司發展。

16.2 Metafont 字體

Knuth 設計了八十多種字體, 其中有些是一般的英文字母, 其餘是用於排版數學符號。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 點之打字機字體 Computer Modern Typewriter 檔名為 cmtt10.mf。Metafont 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。

16.2.1 描邊字型與描點字型

\TeX 所使用之字型大體而言有三個來源: 一是 Metafont 字型, 二是 Type 1 字型, 第三是 TrueType 字型。這三種都是所謂的描邊字型 (outline font)。另外一種字型格式檔是描點字型 (bitmap 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組成; 描點字型是由一個一個的小黑點所構成。當 YAP 預覽軟體將排版結果呈現於顯示器上時, 它先由描邊字型產生圖下方之描點字型。



排版時, \TeX 僅需要每一個字母之寬度、高度等資訊, 但並不須知道字母實際的描繪方式。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。在 \TeX 系統下, 描點字型檔之附加檔名為 .pk, 譬如 cmr12 描點字型檔名為 cmr12.pk。在 \MiKTeX 系統下, 啟動 YAP 預覽/列印軟體時, 顯示器有時會出現 “YAP is creating cmtt10.pk...” 訊息, 表示正在產生描點字型檔。

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。

16.2.2 PostScript 格式之 CM 字體

Type 1 字體也是描邊字型, 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 轉換成 PostScript 格式, 其內儲存文稿所使

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 16.1: 英文與數學字體之比較

<code>\ctxfbb</code>	—	粗黑體
<code>\ctxfbbs</code>	—	斜粗黑體
<code>\ctxfbbe</code>	—	狹長粗黑體
<code>\ctxfbbses</code>	—	狹長斜粗黑體
<code>\ctxfms</code>	—	斜明體
<code>\ctxfme</code>	—	狹長明體

圖 16.2: 中文字體之變形

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。

有一些 Type 1 字體是可以免費使用的, 譬如, 德國 URW++ 公司即捐贈一整套英文字體, 品質相當不錯。但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 Type 1 字體很少。有些人排版一般文字時使用 Type 1 字體, 但數學式部分卻使用 *Metafont* 數學字體。但是, 每一種字體有其風格, 如果你使用 Type 1 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。幸運的是, 目前已有好幾套排版數學式之 Type 1 字體免費提供使用, 16.1 節已略作介紹。

因為一般之 Type 1 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, 至少有兩套免費使用之 PostScript 格式的 CM 字體, 其中一套是由美國數學學會提供。由此可知, 若以 CM 字體排版, 我們可使用 *Metafont* 格式之字型檔, 也可使用 Type 1 格式。MiKTeX 系統會自動處理 CM 字型檔之使用細節, 使用者不須再作設定。

16.3 中文 Type 1 字體

cwTeX 系統所使用的中文字體都是 PostScript 格式。一般的英文 Type 1 字體, 如 Times New Roman, 包含標準字體、斜體、粗體、與粗斜體共計 4 個檔案。cwTeX 中文字體並無類似的字體組合, 但每一種 Type 1 字體都可以作傾斜或水平縮放之變形。因此, 由一種中文字體可以產生多種變化字體出來。譬如, 以中文粗黑體為基礎, 我們還可以使用三種變化字體: 斜粗黑體, 狹長粗黑體, 狹長斜粗黑體。中文字體之變形, 請見圖 16.2。

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%。

16.3.1 中文變形字

中文斜體或狹長字體都是由標準字體變形而來。安裝 $\text{cw}\TeX$ 時，安裝程式會自動設定。斜體字之傾斜度為 16.7 度，狹長字則是將標準字體水平縮小 91%。以上之設定值都可以更改，以下簡單說明。

首先，我們說明字型度量 (font metrics) 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 所使用之字型檔都有一個字型度量檔案，附加檔名為 `.tfm`。譬如，英文 `cmr12` 字型度量檔名為 `cmr12.tfm`；中文字型檔 `m2` 之字型度量檔名為 `m2.tfm`。排版結果如果是輸出為 PS 檔案，則 `dvips` 程式需要 Type 1 字型之度量檔，其附加檔名為 `.afm`。譬如，`m2` 之 Type 1 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。

由此可知， $\mathbb{E}\TeX$ 排版時需要 `.tfm` 檔案，`dvips` 程式輸出 PS 檔案時，需要 `.afm` 檔案，以及描邊字型檔 `.pfb`。以上三種字型檔案之儲存位置如下：

```
.pfb c:\texmf\fonts\type1\cwtex
.afm c:\texmf\fonts\afm\cwtex
.tfm c:\texmf\fonts\tfm\cwtex
```

以明體字 `m2` 為例，`.tfm` 字型度量檔除了上列之標準字體 `m2.tfm` 之外，還有斜體之 `m2s.tfm`，狹長體之 `m2e.tfm`，以及狹長斜體之 `m2es.tfm`。要設定中文變形字之傾斜角度或者水平縮放大小，後三個 `.tfm` 檔案須重新創造。此外，我們還修改字型對應檔之內容，讓 `dvips` 能輸出正確的中文變形字。

中文斜體字內定之傾斜度為 16.7 度，狹長字則水平縮小 91%。以下的例子假設傾斜度為改為 13.0 度，水平縮小比率要改為 0.88。`.tfm` 字型度量檔可以由

PostScript 之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.tfm` 字型度量檔，請先進入 `.afm` 檔案夾內，執行：

```
c:\texmf\fonts\afm\cwtex>afm2tfm m2 -s 0.13 ms2
```

即由 `m2.afm` 產生 `ms2.tfm`，其中，`-s 0.13` 設定傾斜度為 0.13。

同理，如果水平縮小比率要改為 0.88，指令如下：

```
c:\texmf\fonts\afm\cwtex>afm2tfm m2 -e 0.88 me2
```

以上兩種變形同時更改，指令為：

```
c:\texmf\fonts\afm\cwtex>afm2tfm m2 -e 0.88 -s 0.13 mes2
```

請注意，狹長體之字型度量檔名為 `me2.tfm`，斜狹長體之度量檔名為 `mes2.tfm`。每一個中文字型都有 52 個字型度量檔 (`m0.tfm` 至 `m51.tfm`)，依以上方法產生所有的新度量檔後，將它們全部移入 `.tfm` 檔案夾內。

16.3.2 字型對應檔

以上之字型對應檔僅改變 \TeX 之排版版面，`dvips` 程式輸出 PS 檔案之過程也須調整。在 `c:\texmf\dvips\cwtex` 檔案夾內可找到 `cwtex1.map` 字型對應檔，以文字編輯軟體叫出，其內容如下：

```
ms0 CWTEX-M0 ".167 SlantFont" <m0.pfb
ms1 CWTEX-M1 ".167 SlantFont" <m1.pfb
...
me0 CWTEX-M0 ".91 ExtendFont" <m0.pfb
me1 CWTEX-M1 ".91 ExtendFont" <m1.pfb
...
mes0 CWTEX-M0 ".167 SlantFont .91 ExtendFont" <m0.pfb
mes1 CWTEX-M1 ".167 SlantFont .91 ExtendFont" <m1.pfb
...
```

第一行指令之意義為：斜明體字 `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。其他各行的意義應不難理解。要將傾斜度改為 0.13 度，則檔案內的 `.167` 應全部改為 `.13`。同理，水平縮小要變成 0.88，檔案內的 `.91` 應全部改為 `.88`。

其他之中文字型,如粗黑體或仿宋體,也可以作同樣修改,而且每一種字型之設定可以不同。經過以上設定, \ctxfms 可用以排出明斜體字,傾斜度變成 0.13, \ctxfme 則排出水平縮小 88% 之狹長體。

16.4 創造英文 Type 1 字體巨集套件

許多的英文繪圖軟體都附有 Type 1 字體,譬如,英文版 Corel Draw 即附有數百種 Type 1 與 TrueType 字體,Adobe Illustrator 軟體亦附有多種字體。Ghostscript 附有德國 URW++ 公司所捐贈的數十套字型檔案。取得字型檔後,我們須透過巨集套件才能使用這些字體。若 CTAN 上已有現成之巨集套件,我們可直接使用,否則須自行創造。以下以 URW++ 之 Palatino 及 Helvetica 兩套字體為例,簡單說明創造 PostScript 巨集套件的方法。

16.4.1 fontinst 巨集套件

創造 Type 1 字體巨集套件可使用 Alan Jeffrey 與 Rowland McDonnell 所寫的 fontinst 巨集套件。一套英文 Type 1 字體通常包含四個 .pfb 字型檔案,分別對應標準體、斜體、粗體與粗斜體字。若電腦內已安裝 Ghostscript 8.51 版,則 URW++ 所捐贈之字型檔案,包括 .pfb 及 .afm,應置於 c:\Program Files\gs\fonts 檔案夾內。我們首先說明如何創造 Palatino 字體之巨集套件。

Palatino 字體含有 4 個描邊字型檔,檔名如下:

p0520031.pfb (標準體)

p0520231.pfb (斜體)

p0520041.pfb (粗體)

p0520241.pfb (粗斜體)

請注意,檔名最後一個字為英文字母 l,而非阿拉伯數字 1。Ghostscript 檔案夾下可找到 .afm 字型度量檔,但其他繪圖軟體所附字型不見得含有字型度量檔;故底下簡單說明如何自行產生字型度量檔。

在 c:\Program Files\gs\gs8.51\lib 檔案夾內有 pf2afm.bat 檔案,以文字編輯軟體開啓,將以 %GSC% -q 開頭的那一行指令更改如下:

```
c:\Progra~1\gs\gs8.51\bin\gswin32c -q -sDEVICE=nullpage -- pf2afm.ps %1
```

為了方便起見,將此批次檔與上述 4 個 Type 1 字型檔都複製於 c:\xtemp 檔案夾內執行下列指令:

```
c:\xtemp>pf2afm p0520031
```

硬碟內即產生 p0520031.afm。依同法，產生其他 3 個字型度量檔案。程式執行時，會出現下列訊息：

```
pfm file not found -- ignored
```

這不會影響執行結果。

下一個步驟是須更改檔名。為了避免混淆，數以千計的 Type 1 字體必須有一套命名方法。目前通用的命名方法是由 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。許多字體設計公司都提供類似 Palatino 之字體，因此字體名稱上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。有關於字體之命名方法，請見 \texmf\fontname 檔案夾內之說明檔。

依以上原則，Palatino 字體的 4 個字型度量檔案應分別改名如下：

```
p0520031.afm ==> uplr8a.afm
p0520231.afm ==> uplri8a.afm
p0520041.afm ==> uplb8a.afm
p0520241.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，我們利用 fontinst 產生使用字體之巨集套件。進入 DOS，再進入 c:\xtemp 子目錄內，鍵入下列指令：

```
c:\xtemp>tex fontinst.sty
```

當顯示器上出現 * 號，請鍵入：

```
\latinfamily{upl}{}\bye
```

程式執行之後，硬碟中將出現三組檔案，附加檔名分別為 .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。我們建議改用 emTeX 所提供者，原因是執行較方便。emTeX 是 MSDOS 系統下令人懷

念的、古老的 (或者古典的) \TeX 系統, 作者是 Eberhard Mattes。以上兩個檔案已儲存於 `c:\texmf\cwtex\util` 之下。請將檔案複製至 `c:\xtemp` 內, 依序鍵入下列兩道指令:

```
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, 其中 `.vf` 代表 virtual font, 其功能是把字型檔內之字元重新對應。請先在硬碟中的 `c:\localtexmf\fonts` 檔案夾下創造 `\tfm\palatino` 子檔案夾, 將所有的 `.tfm` 檔案移入其內。同樣的, 創造 `\vf\palatino` 子檔案夾, 將所有的 `.vf` 移入其內。

在 `c:\xtemp` 檔案夾內另可找到 4 個 `.fd` 檔案, 其中 `OT1upl.fd`, `T1upl.fd` 與 `TS1upd.fd` 3 個字體驅動檔案, 以文字編輯軟體開啓 `OT1upl.fd`, 其中, 每一行指令都被拆解為 3–4 行。為方便修改, 將多行指令銜接在一起之後, 其內容如下:

```
...
\DeclareFontFamily{OT1}{upl}{}
\DeclareFontShape{OT1}{upl}{b}{n}{ <-> uplrc7t}{}
\DeclareFontShape{OT1}{upl}{b}{sc}{ <-> uplrc7t}{}
...
```

以上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 `uplrc7t` 字型檔排版 Palatino 粗體字, 並使用 OT1 之字元排序方式。

大部分 Type 1 字體之設定尺寸雖然都號稱是 10pt; 但實際大小卻不盡相同。因此, 字體大小須略作調整, 以求搭配。經過幾次嘗試, 我們發現 `OT1upl.fd` 檔案內容應修改如下:

```
...
\DeclareFontFamily{OT1}{upl}{}
\DeclareFontShape{OT1}{upl}{b}{n}{ <-> [.94] uplbc7t}{}
\DeclareFontShape{OT1}{upl}{b}{sc}{ <-> [.94] uplbc7t}{}
...
```

檔案內所有以 `\DeclareFontShape` 指令開頭之設定, 都須加上 `[.94]`, 其作用是讓 `dvips` 在輸出 PS 檔案時, 將字體縮小為原尺寸的 94%。`T1upl.fd` 與 `TS1upd.fd` 請以同樣方法處理。完成以上之修改之後, 請在 `c:\texmf\tex\latex` 檔案夾下創造 `\palatino` 子檔案夾, 再將 4 個 `.fd` 檔案移入其中, 即大功造成。

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
```

倣照處理 Palatino 字型檔的方法, 對 Helvetica 字型檔同樣處理一遍。如此, 硬碟內已存有使用兩種 Type 1 字體所需之檔案。最後, 以文字編輯軟體創造 upl.sty 檔案, 內容如下:

```
%% This is file 'upl.sty',
\renewcommand{\rmdefault}{upl}
\renewcommand{\sfdefault}{uhv}
\endinput
%% End of file 'upl.sty'.
```

第 1 行指令設定 upl 作為標準字體, 第 2 行指令設定 uhv 作為 sans serif 字體。此檔案應儲存於 c:\localtexmf\tex\latex\palatino 檔案夾內。

以上所製作的檔案是供 \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。表 16.3 為字型對應檔案之內容, 檔案取名為 upl.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含有 5 個段落, 其中, 第 1 個段落為 \TeX 之字體名稱。以第 1 行為例, 檔名為 uplr8r, 代表 Palatino 標準字體。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 Type 1 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 16.3 可以看出, 每一字體族須作 6 行設定。前 4 行分別為標準字體、斜體、粗體, 與粗斜體; 第 5 行為數學斜體, 第 6 行為數學粗斜體。

請在硬碟內創造 c:\localtexmf\dvips\upl 檔案夾, 將 upl.map 移入其內。以文字編輯軟體開啓 c:\localtexmf\miktex\config 檔案夾內之 updmap.cfg 檔案, 其中可找到下列設定行:

表 16.3: upl 巨集套件之字型對應檔案

uplr8r	URWPalladioL-Roma "x" <8r.enc <p0520031.pfb
uplri8r	URWPalladioL-Ital "x" <8r.enc <p0520231.pfb
uplb8r	URWPalladioL-Bold "x" <8r.enc <p0520041.pfb
uplbi8r	URWPalladioL-BoldItal "x" <8r.enc <p0520241.pfb
uplro8r	URWPalladioL-Roma ".167 SlantFont x" <8r.enc <p0520031.pfb
uplbo8r	URWPalladioL-Bold ".167 SlantFont x" <8r.enc <p0520041.pfb
uhvr8r	NimbusSanL-Regu "x" <8r.enc <n0190031.pfb
uhvri8r	NimbusSanL-ReguItal "x" <8r.enc <n0190231.pfb
uhvb8r	NimbusSanL-Bold "x" <8r.enc <n0190041.pfb
uhvbi8r	NimbusSanL-BoldItal "x" <8r.enc <n0190241.pfb
uhvro8r	NimbusSanL-Regu ".167 SlantFont x" <8r.enc <n0190031.pfb
uhvbo8r	NimbusSanL-Bold ".167 SlantFont x" <8r.enc <n0190041.pfb

表中之 "x" 應以 "TeXBase1Encoding ReEncodeFont" 字串替代。

```
Map cwtex1.map      #texmf\dvips\cwtex\cwtex1.map
Map cwtex2.map      #texmf\dvips\cwtex\cwtex2.map
```

請在其下加入:

```
Map upl.map        #texmf\dvips\upl\upl.map
```

所有設定即大功告成。最後,我們須更新檔案資料庫。請進入 DOS,依序鍵入下列指令:

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

以上的過程會在 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 upl 巨集套件排版,正文將以 Palatino 標準字體排版。文稿排版結果轉換為 PostScript 格式時, dvips 程式由字型對應檔發現文稿使用 uplr8r 字體, dvips 由 upl.map 檔案之設定得知, uplr8r 字體之對應檔案為 p0520031.pfb 描邊字型檔,即由硬碟取得檔案,轉換出 PS。

16.4.2 選用字體之指令

有了使用 Type 1 字體之巨集套件後,排版時只須在全文設定區引用巨集套件:

```
\usepackage{upl}
```

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章之字體。若只要改變文稿中某一段落之字體, 則全文設定區不應引入巨集套件, 而是在文稿內之段落直接使用下列指令:

```
{\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。此外, 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點數與行距。

字形指令 `\fontshape` 可使用 4 個選項: `n`, `it`, `sl`, 與 `sc`; 其中, `n` 為標準字體, `it` 為斜體, `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` (ultracondensed) 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。

上文說明, Bitstream 公司捐贈一套 Type 1 字體, 名為 Charter。欲使用此一字體, 應在全文設定區加入

```
\usepackage{charter}
```

反之, 如果只是在某一段落使用 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 text}
```

即選用 Charter 字體族, 粗體系列, 標準寬度, 及 small capital 字型。

16.4.3 TrueType 字型

Windows 作業系統內附有一些英文 TrueType 字型, 例如 Times New Roman, Arial 等。pdf \LaTeX 可直接取用 TrueType 字型檔。若 \LaTeX 要使用 TrueType 字型, 方法之一是利用 `ttf2pfb` 將 TrueType 字型檔轉換為 Type 1 字型檔, 再利用 `fontinst` 產生所需之巨集套件。MiK \TeX 系統內含有此一軟體。有興趣嘗試者, 可至 Free Type 網站取得進一步資訊:

`http://www.freetype.org`

17 版面設計

書籍或長篇文稿排版品質的好壞，版面設計是重要關鍵。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計，但修改彈性很小。若某書是以 L^AT_EX 內定之格式排版，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4套巨集套件。第1套是 `geometry`，用於設定版面尺寸。第2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 99) 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。

第3套是同一作者寫的 `titletoc` 巨集套件，用以排版目次。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。第 19 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

如果是排版論文，通常我們只須選定紙張，如 `a4paper`，L^AT_EX 會自動選擇版面之尺寸。但如果是排版書籍，問題就複雜多了。我們必須自行選定紙張尺寸及文字版面之大小。巨集套件 `geometry` 的功能是設定版面之尺寸，作者為 Hideo Umeki。在介紹指令之前，先簡單說明幾個概念。從列印的角度來看，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，`geometry` 巨集套件設定正文方塊尺寸之指令是 `textwidth` 與 `textheight`。以上兩道設定指令對應 L^AT_EX 之 `\textwidth` 與 `\textheight`。但請注意，設定指令之前端並無反斜線。相對於 `textwidth` 指令，`width` 指令為 `textwidth` 加上邊註 (`marginal notes`) 寬度，以及邊註與文字版面之間距。

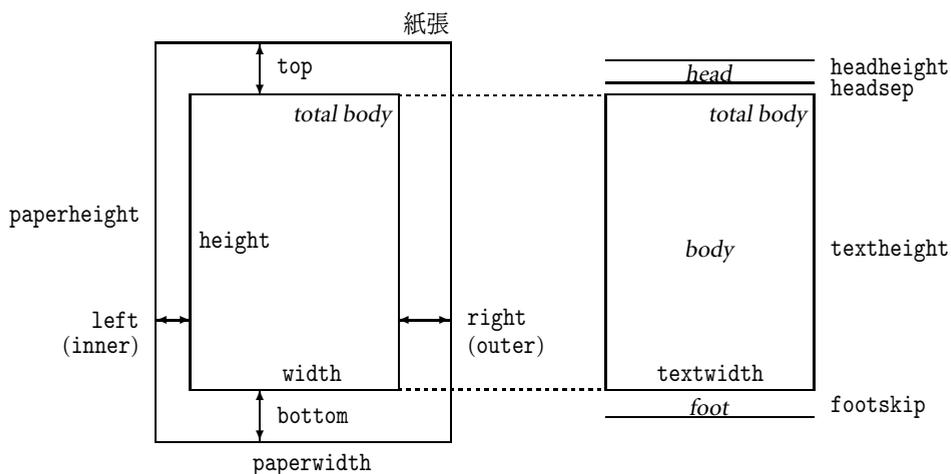


圖 17.1: geometry 巨集套件設定指令圖示

說明: 此圖取自 geometry 巨集套件 (v. 3.2) 說明檔。

根據圖 17.1, 若要以 geometry 巨集套件 (v. 3.2) 指令設定正文方塊之尺寸, 僅須在全文設定區加入下列一行指令:

```
\usepackage[textwidth=17cm, textheight=25cm]{geometry}
```

如果要進一步設定左邊緣為 2.5 公分, 上邊緣為 3.5 公分, 設定指令為:

```
\usepackage[textwidth=17cm, textheight=25cm,
left=2.5cm, top=3.5cm, 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, 故以上第一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 紙張, 應加入 a4paper 設定項。或者, a4paper 也可以直接加入於 \documentclass 選項內。

除了圖 17.1 所列出的指令外, geometry 巨集套件還有許多其他的設定指令, 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, 左右邊沿各留下 1 吋空白, 指令為:

表 17.1: geometry 巨集套件選項

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。
asymmetric	有些書籍之單雙頁採用相同之版面, 此時可以 asymmetric 選項排版。
hoffset	同 T _E X 之 \hoffset 指令。
voffset	同 T _E X 之 \voffset 指令。

```
\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設定內容加於 \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, 長度是以英吋為單位, 但我們也可以下列方式設定:

```
\usepackage[hmargin{0.1\textwidth,0.1\textwidth}]{geometry}
```

表 17.1 列出一些較常用的設定指令。本書之設定如下:

```
\geometry{paperheight=25.5cm,paperwidth=17cm,textwidth=11.5cm,
textheight=20.5cm,top=1.5cm,footskip=1.5cm,
marginparwidth=2.5cm,left=2cm,twoside}
```

紙張寬度是 17 公分, 文字版面寬度則是 11.5 公分。指令最後的 twoside 是用以調整單雙頁之版面位置。排版書籍時, 常見的作法是, 單雙頁之版面採不同的設計, 以上之 twoside 指令即作此用途。7.1 節曾說明, 我們可使用 \hoffset 與 \voffset 指令控制文字版面在紙張上之列印位置。若要以 geometry 指令設定, 則使用 hoffset 與 voffset 選項控制。

17.2 chngpage 巨集套件

排版書籍時，正文的行長以 `geometry` 設定為 12 公分，書末的索引可能爲了節省紙張，文字版面寬度可能加大爲 14 公分。或者，教科書每一章後面之習題若採兩欄排版，文字版面寬度也須加大。行文當中欲加入版面寬度，可使用 Peter Wilson 所寫的 `chngpage` 巨集套件。

最基本的指令如下：

```
\changetext{textheight}{textwidth}{evensidemargin}{oddsidemargin}{columnsep}
```

本指令計有 5 個選項，各選項之意義請見圖 7.1 (頁 85)。選項若留爲空白，則採原設定值，若填入數字 (及單位)，此數值即加入原設定值。例如，若排版索引之前加入以下指令：

```
\changetext{}{2cm}{-2cm}{}{}
```

對照以上之定義，`textheight` (文字版面高度) 不變，`textwidth` (文字版面寬度) 加大 2 公分，`evensidemargin` (偶數頁左邊) 減 2 公分，`oddsidemargin` (奇數頁左邊) 維持不變。最後，若採兩欄式編排，`columnsep` (欄距) 維持不變。行長加大 2 公分時，偶數頁頁面須往左移，否則與原版面無法對齊，這是第 3 個選項之作用。

排版索引之後，版面要回復原始設定值，指令如下：

```
\changetext{}{-2cm}{2cm}{}{}
```

17.3 titlesec 巨集套件進階指令

就書籍排版而言，章節標題之設計是很重要的一部分。但遺憾的是， \LaTeX 之章節標題指令調整彈性很小。欲自行設計章節標題，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 (頁 99) 曾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，前者用於設計章節標題式樣，後者則用於控制間距。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。爲求文稿風格統一，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` 指令也可以加在特定章節標題之前。章節標題可分為兩部分: 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。

巨集套件 `titlesec` 之指令相當複雜, 我們將較重要的指令列於表 17.2。由表中第一行可知, 使用 `\titlesec` 指令時須加入 7 個選項。以圖 17.2 為例, 各選項之功能說明如下。

- 第 1 選項 *command* 選擇欲更改設計之標題。圖 17.2 的例子選用 `\chapter`, 表示要設計章之標題。如果要更改節標題, 應填入 `\section`。
- 第 2 選項 *shape* 設定章節編號與標題文字之排版方式。圖 17.2 的例子使用 `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版。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, 此處之指令對於章節編號與標題文字都有作用。圖 17.2 的例子使用 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。圖 17.2 例子編號部分為「第 17 章」格式, 並選用中文圓體字。其中, 章編號是以 `\thechapter` 指令輸入。排版各章之標題時, 我們以 `\chapter` 指令輸入標題文字, 其字體之設定是利用 `\ctxfdef` 設定:

```
\ctxfdef{\chapter}{\ctxfm}{\ctxfr}
```

- 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。圖 17.2 的例子中, 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。

表 17.2: 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*: 編號與標題文字在同一行 (內定值)
 - *block*: 編號與標題文字合在一起為一段落
 - *display*: 編號與標題文字各成一獨立段落
 - *runin*: 編號與標題文字在同一後, 其後接正文
 - *leftmargin*: 編號與標題文字排在版面左沿上
 - *rightmargin*: 編號與標題文字排在版面右沿上
 - *drop*: 標題嵌入文字段落內
 - *wrap*: 同上, 但標題文字長度會自動調整
 - *frame*: 同 *display*, 但加上外框
- *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*: 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。
- *aftersep*: 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 17 章

版面設計

```
\documentclass[11pt,a4paper]{book}
\usepackage[sf]{titlesec}
\titleformat{\chapter}[display]{\centering\Large\sf}
  {\ctxfr 第~\thechapter~章}{0.2cm}{}
\ctxdef{\chapter}{\ctxfrm}{\ctxfr}
...
\chapter{版面設計}
```

圖 17.2: 章標題設計

圖 17.2 的例子以 book 文稿類別排版，標題要居中編排，章編號以「第17章」格式出現。第 2 行引入 `titlesec` 巨集套件，因為加入 `sf` 選項，故節與小節之標題雖然未重新設計，但將改用 `sans serif` 字體。第 3–4 行是 `\titleformat` 設計章標題之排版，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，在 `titlesec` 巨集指令內，註銷符號 `%` 可以省略不加。

本例之章編號以阿拉伯數字編排，有人可能偏好以國字數字編排。若第 4 選項之內容改為：

```
{\ctxfr 第十七章}
```

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，下一章之標題編號仍然排為「第十七章」。如果不怕麻煩，我們可在每一章 `\chapter` 指令之前重新定義 `\titleformat` 指令，即可排出正確之國字編號。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。

17.3.1 彩色標題

標題文字常以彩色排版，本小節說明排版方法。`\titleformat` 指令的第 6 選項主要是控制標題文字。排版章節標題時，我們可以使用 `color` 巨集套件之 `\textcolor` 指令將文字段落變成彩色，譬如，

```
\chapter{\textcolor{blue}{版面設計}}
```

第 17 章

版面設計

```
\documentclass[11pt,a4paper]{book}
\usepackage{titlesec,color}
\newcommand{\mytitle}[1]{\textcolor{blue}{#1}}
\titleformat{\chapter}[display]{\centering\Large\sc}
  {\ctxff \textcolor{blue}{第~\thechapter~章}}{0.2cm}{\mytitle}
\ctxfdef{\chapter}[\ctxfm]{\ctxff}
...
\chapter{版面設計}
```

圖 17.3: 彩色標題

以上之指令必須加在每一章之 `\chapter` 指令內，並不方便。而且，如果原先選定某一顏色，後來改變主意，則每一章之指令都須更改，也很麻煩。

`titlesec` 巨集套件提供的解決辦法如下：我們可定義一內含章節標題文字之巨集指令。將此巨集指令引入 `\titleformat` 指令選項內時，即可加入指令以控制標題之排版。圖 17.3 即以此方法將標題上色。首先，第 3 行指令定義一巨集指令稱為 `\mytitle`，其中所包含之 `#1` 即代表各章之標題。換言之，當我們輸入 `\chapter{版面設計}` 指令時，「版面設計」4 個字即代入 `#1`。第 3 行指令把 `\textcolor{blue}` 指令加在 `#1` 上，故各章之標題文字將以藍色排版。

以上之指令僅將章標題文字轉為藍色，編號部分仍以黑色排出。如果章編號也要以藍色排版，必須另作處理。`\titleformat` 控制章節編號之指令是置於第 4 選項內。圖 17.3 之例子中，第 4 選項已改為：

```
{\ctxff \textcolor{blue}{第~\thechapter~章}}
```

故編號部分也是以藍色排版，中文字體則改用仿宋體。

依 \TeX 之內定值，章編號與標題文字是分為兩行排版，此稱為 `display` 格式。圖 17.4 之例子則採用 `hang` 格式，編號與標題文字將排於同一行，這也是 \TeX 排版節標題 `\section` 之格式。指令第 3 行引入 `xcolor` 巨集套件以及 `dvipsnames` 選項，以方便使用顏色。本例把章編號及標題文字全部納入 `\mytitle` 指令內。為了加上灰階之背景方塊，我們把編號與文字置於 `\parbox` 段落方塊內，再以 `\colorbox` 指令包以 `lightgray` 之灰階色（第 4 行）。

排版之後，我們希望灰階方塊與文字版面同寬，這需要作一點計算。依指令第 6 行 `\fboxsep` 之設定，灰階色外框與 `\parbox` 之間距為 `0.05\textwidth`，故

17 版面設計

```
\documentclass[11pt,a4paper]{book}
\usepackage{titlesec}
\usepackage[dvipsnames]{xcolor}
\newcommand{\mytitle}[1]{\colorbox{lightgray}
  {\parbox{.9\textwidth}{\textcolor{blue}{\thechapter ~ #1}}}}
\titleformat{\chapter}[hang]{fboxsep=.05\textwidth\sc\huge}
  {}{0mm}{\mytitle}
\ctxdef{\chapter}{\ctxfm}{\ctxff}
...
\chapter{版面設計}
```

圖 17.4: 單行標題

灰階方塊兩邊合計占用了 0.1\textwidth 。因此，我們把段落方塊 `\parbox` 之寬度設定為 0.9\textwidth ，合計即與文字版面同寬。

再舉另外一個以 `hang` 格式排版之例子，本書排版章標題之指令圖 17.5 所示。首先，指令第 2 行以 `\fontfamily` 指令定義英文字體 `\chfont`，此字體是 `mathpazo` 巨集套件之英數字體，字體族之名稱為 `ppl`。每一章首頁之章標題占有一定空間，若嫌空間太大或太小，可自行調整。`\titleformat` 指令之第 3 選項內加入 `\vspace*{-5mm}` 將標題位置提高 0.5 公分。

第 4 選項排版章編號時，使用 `\chfont` 字體，並以 0.4 深淺度之灰階色 `heavy`。章編號之位置是先以 `\rule` 指令由文字版面左沿右移 0.4\textwidth 之距離，再以 \TeX 之 `\llap` 指令讓編號數字 `\thechapter` 往左編排，亦即，數字之右沿將切齊此點。此一作法可保證各章之文字標題是從版面上同一位置往右編排。接下來，編號與標題文字之間距設定為 0.4 公分；並使用 `\huge` 字級。最後一個選項是 `\noindent`，其作用是讓章標題下之文字從版面左沿開始排版，不要內縮。

```
\definecolor{heavy}{gray}{.4}
\newcommand{\chfont}{\fontfamily{ppl}\fontseries{m}\fontshape{sc}%
  \fontsize{20}{25pt}\selectfont}
\titleformat{\chapter}[hang]{\vspace*{-5mm}}
  {\textcolor{heavy}{\chfont\rule{.4\textwidth}{0pt}\llap{\thechapter}}}
  {4mm}{\CXtitle}[\noindent]
```

圖 17.5: 章標題

17.3.2 章節計數器

使用 \LaTeX 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 $\text{\backslash thechapter}$ 代表，其來源是 \chapter 計數器。因此，若本章為第 17 章， \LaTeX 自 \chapter 計數器中取出“17”，設定以阿拉伯數字格式編排。若章編號要改為 20，我們須在 \chapter 指令之前重新設定計數器之數值：

```
\setcounter{chapter}{19}
```

排版時， \LaTeX 碰到下一個 \chapter 指令，會先把計數器加 1，再取出其中的數字排版章編號。

17.3.3 節標題之設計

上一小節提供幾個排版章標題的例子，本小節則說明節標題之設計。事實上，這些指令也可以應用於排版小節標題或者章標題。如果是排版段落 \paragraph 或次小節 \subsubsection 之標題，我們可以進一步變化其設計。

依 \LaTeX 之原始設計，章標題之編號與文字標題分兩行編排；節標題則是兩者排於同一行。圖 17.6 之例子把節標題分為兩行，第 1 行排版數字編號，第 2 行為標題文字。 \titleformat 指令第 2 選項為 \display ，指示節編號與標題文字各成一個段落。最後的第 6 選項內之指令是用於排版標題文字。本例中，我們使用 \TeX 之 \hrulefill 指令畫一橫線，其前之 $\text{\vspace*{-15pt}}$ 指令用以縮減橫線與其上文字段落之間距；橫線指令之後加上換行指令，並加大行距 0.1 公分。

圖 17.7 是另一個節或小節標題的例子。 \titleformat 指令的第 2 個選項選用 \runin 設定，在洋文書排版術語中，這稱為 \running-head ：節或小節之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之排版。本例中，此選項內有兩道指令， \normalfont 指令將英文與數字回復標準字體， \sffamily 則選用 \sans serif 字體。第 4 選項是控制節編號之排版，本例之指令為：

```
{\S\ \thesection}
```

我們在節數字編號之前加上一個裝飾符號 \S ，其後空一格，之後再加上節編號 \thesection 。以某書之 17.3 節為例，數字編號將排為： \S 17.3 。

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 0.5cm。第 6 選項內是用於控制標題文字。

第 17.3 節

章節標題設計

書籍與長篇文稿通常區分章節，章節標題之編排設計必須前後一致。L^AT_EX 提供各種編排章節標題之之指令，這些指令已能滿足多數人的需求。

```
\documentclass[11pt,a4paper]{book}
\usepackage{titlesec}
\ctxfdef{\section}{\ctxfm}{\ctxff}
\titleformat{\section}[display]{\large}
  {\ctxff 第~\thesection~節}{3mm}
  {\vspace*{-15pt}\hrulefill\{1mm}}
...
\fontsize{10.95}{18pt}\selectfont
\chapter{版面設計}
\section{章節標題設計}
書籍與長篇文稿通常區分章節， 章節標題之編排設計必須前後一致。
LATEX 提供各種編排章節標題之之指令， 這些指令已能滿足多數人的需求。
```

圖 17.6: 節標題

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 [---]，因此標題文字之後加上一短橫線。

17.3.4 嵌入標題

本小節介紹西文排版中所謂的 drop 標題，本書稱之為「嵌入標題」。嵌入標題之設計通常用於小節或次小節，標題並非單獨成一行，而是直接置於節段落前端之左沿。上小一節之 running-head 標題也是直接置於段落文字前端，兩者不同的地方在於，running-head 之標題文字排為一行。相對而言，drop 標題文字之設計如圖 17.8 所示。小節標題之編號與文字可能占用 2–3 行的高度，標題本身自成一段落小方塊，排版於小節前端，標題文字可能排為 2 行以上。

在 titlesec 巨集套件中，我們可以用 drop 或 wrap 選項排版嵌入標題，圖 17.8 的例子使用 wrap 選項。titlesec 指令的第 2 選項 wrap 選用嵌入標題。第 3 選項主要選用粗黑字體，但最後一個指令為 \filright；這是 titlesec 巨集套件之指令，其功能與 L^AT_EX 之 \raggedright (靠左編排) 類似，但取消英文單字之音節斷字 (hyphenation)。因為標題占用一小空間，如果英文單字再使用音節斷字，排版效果不好。不過，如果是排版中文標題，就沒有音節斷字的問題。

§ 17.3 章節標題設計— 書籍與長篇文稿通常區分章節，章節標題之編排設計必須前後一致。L^AT_EX 提供各種編排章節標題之之指令，這些指令已能滿足多數人的需求。

```
\documentclass[11pt,a4paper]{book}
\usepackage{titlesec}
\ctxfdef{\section}{\ctxfm}{\ctxfr}
\titleformat{\section}[runin]{\normalfont\sffamily}
  {\S \thesection}{.5cm}{[---]}
...
\fontsize{10.95}{18pt}\selectfont
\chapter{版面設計}
\section{章節標題設計}
書籍與長篇文稿通常區分章節，章節標題之編排設計必須前後一致。
LATEX 提供各種編排章節標題之之指令，這些指令已能滿足多數人的需求。
```

圖 17.7: running-head 標題

使用 `wrap` 選項時，我們須加入 `\titlespacing` 指令，其第 2 選項設定標題文字所占之寬度；本例之寬度為 3 公分。寬度決定之後，各節標題文字的長度不一，若設定之長度小於標題文字之長度，輸入標題文字時，我們須自行加上換行指令。本例之標題文字即分為兩行。`\titlespacing` 指令的第 3 選項與第 4 選項分別設定標題文字前後之間距，本例分別選用 0.3 公分與 0.5 公分。

17.3.5 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，因此其標題之編排須另行處理。

L^AT_EX 提供 `\tableofcontents` 指令以排版目錄。依內定值，目錄之標題是以 `\chapter*` 指令編排，* 號表示不編號。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。如果正文各章是以 `\titlesec` 設計標題格式，而序文與目錄之標題以 `\chapter*` 排版，則後者將以 L^AT_EX 內定之格式排版，兩者可能不太搭配。

本書之作法如下。我們定義一巨集指令，取名為 `\numtitle`，以之排版各章之標題，其中使用 `\titlesec` 指令設計標題。相對的，序文與目錄之標題並無編號，故我們另外定義一巨集指令，名為 `\nonumtitle`，其中仍以 `\titlesec` 指令設計標題，但不加入編號。排版各章標題之前，我們加上 `\numtitle` 指令以控制標題

17.3 章節標題 書籍與長篇文稿通常區分章節，章節標題之編排設計必須與排版須前後一致。L^AT_EX 提供各種編排章節標題之之指令，這些指令已能滿足多數人的需求。但是，如果你有較特別的設定需求，必須使用其他巨集套件。

```
\documentclass[11pt,a4paper]{book}
\usepackage{titlesec}
\ctxdef{\section}{\ctxfm}{\ctxff}
\titleformat{\section}[wrap]
  {\normalfont\fontseries{b}\selectfont\filright}{\thesection}{.6em}{
\titlespacing{\section}{3cm}{3mm}{5mm}
...
\fontsize{10.95}{18pt}\selectfont
\chapter{版面設計}
\section{章節標題\ 與排版}
書籍與長篇文稿通常區分章節，章節標題之編排設計必須前後一致。
LATEX 提供各種編排章節標題之之指令，這些指令已能滿足多數人的需求。
但是，如果你有較特別的設定需求，必須使用其他巨集套件。
```

圖 17.8: 嵌入標題

設計，而排版序文與目錄之前，我們則加上 `\nonumtitle` 指令。

以 `\chapter*` 指令排版目錄或序文，還有另一個問題要解決：以 `\chapter*` 指令所排版的標題不會自動納入目錄內。欲解決此一問題，我們可在 `\chapter` 指令之前加入下列指令：

```
\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，如果我們以 `\chapter*` 排版序文標題，則指令如下：

```
\addcontentsline{toc}{chapter}{\ctxff 序文}
\chapter*{序文}
```

排版後，「序文」及其頁次將排入目錄內，並以仿宋體編排。

17.4 titletoc 巨集套件

欲排版目錄，可使用 `\tableofcontents` 指令，此一指令通常是下在第 1 章或序文之前。排版時，L^AT_EX 會在目錄上方加上 **Contents** 一字作為標題。如果是中文稿，我們可以在全文設定區加入以下之指令

<div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> ← left right → </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>5 消費函數</p> <p>5.1 跨期的消費選擇</p> <p style="padding-left: 2em;">借貸市場, 132 ▶ 儲蓄, 134, ▶ 預算限制式, 136</p> </div> <div style="width: 5%; text-align: center;"> <p>130</p> <p>131</p> </div> </div>
--

圖 17.9: 目錄圖示

```
\renewcommand{\contentsname}{\ctxfr 目錄}
```

將目錄之標題改為中文圓體。

\LaTeX 自動以 `\chapter*` 指令排版目錄之標題。如果我們是用 `\titlesec` 指令設計正文各章之標題, 上一小節關於排版序文標題之討論也適用於此處。為求版面之一致, 我們應該同時設計一個不含編號之標題以排版目錄標題。本書之目錄是以 `\tableofcontents` 指令排版, 但標題以 `\titlesec` 指令設計。目錄本身占有一定頁數, 而各章之頁碼又會出現在目錄內, 故欲得到正確的頁碼, 我們須執行 `latex` 程式 2-3 次。

如果要自行設計目錄之排版方式, 可使用 `titletoc` 巨集套件, 這是 `titlesec` 作者 Javier Bezos 的另一項創作。`titletoc` 巨集套件提供 `\titlecontents` 指令以控制目錄之格式。使用此一巨集套件時, 我們仍然是以 `\tableofcontents` 指令排版目錄, 但另以 `\titlecontents` 指令控制目錄之排版格式。

介紹指令之前, 我們先了解目錄之一般樣式。圖 17.9 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。章與節目錄之格式大體依照 \LaTeX 內定之格式, 小節目錄則重新設計。圖 17.9 的左右兩條直線代表文字版面邊沿。右沿版面 (*right*) 排版頁碼, 其空間大小可由 `titletoc` 之指令設定, 例如, 右沿版面若占 0.8 公分, 指令為:

```
\contentsmargin{8mm}
```

`\titletoc` 並無指令控制左沿版面, 但其寬度可直接由 `\titlecontents` 指令控制。`\titlecontents` 指令共有 7 個選項以控制目錄之格式, 表 17.3 說明指令之用法。如果你已熟悉 `\titlesec` 指令, 應該不難掌握指令之用法。

舉例言之, 若使用 \LaTeX 之 `book` 文稿類別排版, 目錄中之節標題大約是以下列指令定義:

```
\titlecontents{section}[6mm]{\contentslabel{8mm}}
{\hspace*{-8mm}}{\titlerule*[10pt]{.}\contentspage}
```

表 17.3: titletoc 巨集套件指令

```
\titlecontents{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 17.9。
- *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 17.2, 頁 288) 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。
-

第 1 個選項指示要更改 `\section` 之目錄排版方式。第 2 個選項為方括號內的 6mm, 設定版面左沿 (圖 17.9 之 *left*) 之寬度。第 3 個選項為空白, 表示整個目錄項之格式 (如目錄項之間距) 採內定值。第 4 選項控制節標題文字前之排版。圖 17.9 的例子中, 這是指「跨期的消費選擇」前面之空間; 其後即自動排入節標題文字。本例之指令為 `\contentslabel{8mm}`, 表示編號 “5.1” 及其後的空間共占 0.8 公分。

第 5 選項之功能同第 4 選項, 但適用於沒有編號之節或小節 (如 `\section*`)。若無編號, 則不需占用任何空間, 故本例加上 `\hspace*{-8mm}` 指令左移 0.8 公分, 以與上一個節編號左沿對齊。第 6 個選項以 `\titlerule*` 指令 (參見表 17.3) 設定細點之間距為 10pt, 行末則填入頁碼 `\contentpage`。

圖 17.9 例子的小節目錄較特別，小節不加上編號，目錄文字之後不加上細點直線，而是緊接著排出頁碼。`\titlecontents` 指令之設定如下：

```
\definecolor{half}{gray}{.50}
\newcommand{\triang}{\fboxsep=0pt
\textcolor{half}{\blacktriangleright}}
\titlecontents*{subsection}
[14mm]{\fontsize{10.95}{16pt}\selectfont\vspace*{0.6mm}}
{}{}{, \thecontentspage}[ ~\triang{~} ]
```

兩個小節目錄文字之間是以一右三角形區隔。若直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。因為要使用右三角形的符號，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。

本例之小節目錄不排版編號，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]` 設定版面左沿之寬度，第 3 選項選用字級與行距，節與小節目錄之間距則加大 `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 選項控制，我們先加上逗點，再以 `\thecontentpage` 指令填上頁碼。最後，右三角形符號則加於第 7 個選項內，符號前後各留出一點空白。

`titletoc` 巨集套件還可以作其他排版設計，例如，我們可在每一章的題目之下排版該章之目錄，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，說明檔內提供不少例子；仔細摸索，不難由之設計你所要的格式。

17.5 頁面格式

一般文稿通常在會版面右上角或正下方加入頁碼。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。本書即在雙數頁之左下角排版頁碼與章標題；單數頁之右下角則排版節標題與頁碼。在西文排版中，頁眉與頁足之排版設計稱為頁面格式 (page style)。L^AT_EX 提供 `\pagestyle` 指令以排版頁面格式。若我們僅是要排版出頁碼，使用 `\pagestyle` 指令已足夠。但是，若頁眉與頁足要排版中文章節標題，則須作設定，否則中文部分會出現亂碼。

若要自行設計中文稿之頁面格式，我們可以使用 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 17.10 畫出版面

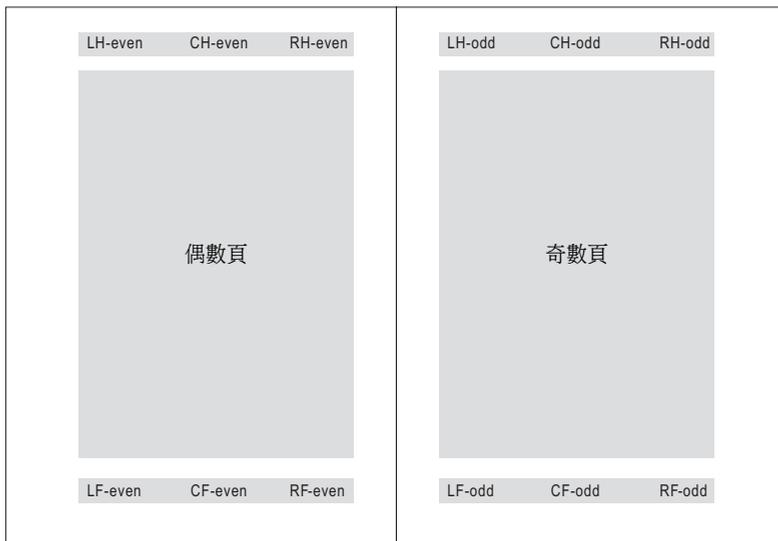


圖 17.10: 頁面格式

上頁眉與頁足之位置。在洋文書的排版中，以 book 文稿類別所排版的書籍，左右頁之版面格式不同。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 (left header)，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 (center footer)。

17.5.1 頁眉之章節標題

\LaTeX 以 `\pagestyle` 指令排版頁面格式 (page style)，此一指令可選用 4 個選項: `empty`, `plain`, `headings`，與 `myheadings`。如果選用 `empty`，則頁眉與頁足全部空白，連頁碼都沒有。若選用 `headings`，頁眉會排版章節標題與頁碼，頁足為空白。章節標題之排版是依 \LaTeX 內定之格式。如果要自行設計頁眉與頁足之排版，可選用 `myheadings`，再以 `\markboth` 與 `\markright` 指令控制排版內容與格式。

根據以上之說明，若在全文設定區加上以下指令：

```
\pagestyle{headings}
```

頁眉將排版出章節標題與頁碼，其內容依文稿類別而定：

- 以 book 文稿類別排版，或者以 article 排版但加入 `twoside` 選項
若是以 book 文稿類別排版，文稿之左右頁對稱於書脊，左頁書眉右邊將排版章標題；左邊則排版頁碼。相對的，右頁書眉左邊將排版節標題，右邊則

排版頁碼。反之，若是以 article 文稿類別排版，左頁眉排版節標題，右頁眉排版小節標題。

- article 文稿類別排版

左右頁之頁面格式相同，頁眉左邊排版節標題，右邊排版頁碼。

以上之指令是在頁眉排出章節標題，但不幸的是，若是中文稿，頁眉之中文標題會變成亂碼。要避免出現亂碼，我們必須在全文設定區作適當設定。

在說明設定方法之前，我們先簡單說明 \LaTeX 如何在頁足與頁眉上排版章節標題。爲了排版章節標題， \LaTeX 必須將每一頁之章節標題記錄下來，其中的細節相當複雜。簡單來說，記錄章節標題之指令是 \LaTeX `\markboth` 與 `\markright`：

```
\markboth{main-mark}{sub-mark}
\markright{sub-mark}
```

以 book 文稿類別而言， \LaTeX `\markboth` 的 *main-mark* 記錄章標題，*sub-mark* 則記錄節標題。相對的， \LaTeX `\markright` 則僅記錄節標題。若是以 article 文稿類別排版， \LaTeX `\markright` 也是記錄節標題。

以上之指令僅記錄章節標題之內容，實際排版是使用下列三個指令：

```
\chaptermark
\sectionmark
\subsectionmark
```

第 1 個指令 \LaTeX `\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。依 \LaTeX 之內定值，頁眉上之章標題除了章名之外，還會自動加上英文 CHAPTER 與數字編號，格式如“CHAPTER 17. 版面設計”。若要去掉章編號之前的英文字變成簡單的“17.”，須在全文設定區加入指令：

```
\renewcommand{\chaptermark}[1]{\markboth{\thechapter.\ #1}{}}
```

以上指令重新定義 \LaTeX `\chaptermark` 設計之頁眉與頁足之章標題格式。 \LaTeX `\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 文稿類別中，第一對大括號內存放欲出現在左頁眉章標題處之文字；第二對大括號內則存放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爲空白。第一對大括號是章標題之設計， \LaTeX `\thechapter` 指令爲章編號，其後緊接一英文句點；再以 \LaTeX `\` 指令加入一空格，最後的 #1 參數則代表章名。經過以上之定義，左頁眉之章標題排版爲：

17. 版面設計

如果頁眉之章編號要改為「第 17 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尙可自行設定。

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編排, 則章標題出現於頁眉時也是粗黑體。若頁眉 (以及目錄內之章名) 要改用其他字體, 輸入 `\chapter` 指令應加入選項, 全文設定區再以 `\ctxfdef` 指令選用中文字體:

```
\ctxfdef[\ctxff]{\ctfbb}  
...  
\chapter[版面設計]{版面設計}
```

經以上設定, \TeX 將以仿宋體排版頁眉/頁足以及目錄內之章名。事實上, 方括號選項內之文字可以和大括號內之標題文字不同。譬如, 若章標題文字甚長, 方括號內可輸入較簡短之標題。

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用類似的指令更改。以節標題為例, 設定指令為 `\sectionmark`。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, 若重新定義 `\subsectionmark` 之內容, 即可變更頁眉/頁足之小節標題之格式。

回到我們原先的問題: 如何避免頁眉之中文出現亂碼? 如果使用 `book` 文稿類別, 設定指令如下:

```
\pagestyle{headings}  
\renewcommand{\chaptermark}[1]{\markboth{\#1}{}}  
\renewcommand{\sectionmark}[1]{\markright{\thesection~ \#1}}
```

以上第 2 行之 `\chaptermark` 代表章標題。如果是使用 `article` 文稿類別排版, 文稿內並無 `\chapter` 指令, 故第 2 行指令應去掉。 \TeX 會自動選用斜體字排版頁眉之標題, 但我們可以選用其他字體。例如, 第 3 行指令若改為

```
\renewcommand{\sectionmark}[1]{\markright{\normalfont\thesection~ \#1}}
```

則以正體字排版。



亂碼!

17.5.2 fancyhdr 巨集套件

以上之指令可以重新設計頁眉或頁足之排版,但指令稍嫌複雜。要自行設計頁眉與頁足之格式,應使用 fancyhdr 巨集套件(作者 Piet van Oostrum)。我們首先說明以 article 文稿類別排版的例子,其單雙頁之版面是相同的。

假設文稿是以 article 文稿類別排版,題目是「排版的原則」。我們希望在頁眉右方排出題目,則全文設定區應加入以下之設定指令:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
\rhead{\ctxff 排版的原則}
\cfoot{\thepage}
\renewcommand{\headrulewidth}{0pt}
```

TEX 之 \pagestyle 指令可選用 empty, plain 等選項, fancyhdr 新增 fancy 選項。第 2 行指令選用 fancy 格式,故接下來我們必須定義 fancy 格式。第 3 行之 \fancyhf{} 指示清除所有之頁眉與頁足。第 4 行以 \rhead 指令定義頁眉右邊要排版「排版的原則」。第 5 行則以 \cfoot 定義頁碼 \thepage 要排於頁足中間。末行之 \headrulewidth 定義頁眉底下橫線之粗細。本例設為 0pt,故不畫橫線。

依 TEX 之內定值,頁眉標題之下會自動加一橫線。最後一行指令是把此一橫線之粗細設為 0,等於是不要橫線。如果要把「排版的原則」5 個字排版於頁眉左邊,則 \rhead 應改為 \lhead。若要排版於頁足右邊,則使用 \rfoot。反之,如果使用以下之指令:

```
\rfoot{\ctxff 排版的原則, ~\thepage }
```

同時,原來的第 5 行指令去掉,則頁足右邊會排出題目,再排版頁碼。

以上的例子是自行輸入頁眉之節標題,但較方便的作法是讓 TEX 自動取用標題文字。我們可以使用 \rightmark 與 \leftmark 指令讓章節標題自動排版於頁眉/頁足上。簡單來說,若以 book 文稿類別排版, \leftmark 是章標題, \rightmark 是節標題。反之,若以 article 文稿類別排版, \leftmark 是節標題, \rightmark 則是小節標題。因此,若以下列指令

```
\fancyhead[RO,RE]{\leftmark}
```

取代上面例子中之第 4 行,則頁眉右邊會自動排出節標題。

根據西文書之排版規範,每一章首頁之頁眉/頁足並不排版章節標題。如果某頁只要排版頁碼,其餘的資訊都不出現,僅須在該頁某處加入以下指令即可:

```
\thispagestyle{plain}
```

不過, 若以 `\chapter` 指令排版章標題, 會自動加入上述指令。另外, 以 `article` 文稿類別排版時, \TeX 也會自動在 `\maketitle` 指令之後加入上述指令, 不須再自行加入。如果標題頁連頁碼都不要, 則應自行加入 `empty` 選項:

```
\thispagestyle{empty}
```

17.5.3 單雙頁版面不同

以 `book` 文稿類別排版時, 單雙頁之版面格式不同, 故單雙頁版面都須定義。 `fancyhdr` 巨集套件所提供之基本的設定指令如下:

```
\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` 則代表奇數頁左欄。

在 `book` 文稿類別下引用 `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 行指令設定奇數頁左欄與偶數頁右欄為章編號與標題。未行之 `\footrulewidth` 定義頁足橫線之粗細。本例為 `0pt`, 故不畫橫線。

相對於以上之版面, 一個簡單的設計如下。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` 格式之定義如下:

```
\fancyhf{}  
\fancyhead[LE]{\leftmark}  
\fancyhead[RO]{\rightmark}  
\fancyhead[RE,LO]{\thepage}  
\headrulewidth=0pt
```

本書之頁眉為空白，頁碼與章節標題全部置於頁足，設計指令於下：

```
\fancyhf{}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection~ #1}}
\fancyfoot[LE]{$\cdots\thepage\cdots\rule{4mm}{0pt}\leftmark}
\fancyfoot[RO]{\rightmark\rule{4mm}{0pt}\cdots\thepage\cdots}
\renewcommand{\headrulewidth}{0pt}
```

以上指令在偶數頁頁足之左欄排版章標題，但不編號。奇數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。

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 文稿類別時，每一章開頭之 `\chapter` 指令出現處， \TeX 自動加入上述指令，因此我們不須再自行輸入。事實上，`plain` 之格式也可以自行調整。譬如，學術期刊內每一篇論文首頁的正下方通常會排版版權聲明。如果我們是以 book 文稿類別排版整本期刊，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識：

© *TER*

第 4-5 行設定頁眉與頁足都不要橫線。

18 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。但是，編製索引的工作非常繁瑣。以往電腦尚未應用於排版時，中文書很少編製索引。L^AT_EX 提供一些輔助工具，包括工具程式 `makeindex`，使編輯索引的工作變得容易許多。重要的是，每一索引項都很精確。

不過，L^AT_EX 的工具程式無法正確處理中文，cW^AT_EX 提供 `cwmkidx` 程式，以方便編製中文索引。除了正確處理中文之外，`cwmkidx` 程式會對中文索引名詞排序，並算出筆劃。

18.1 標識索引名詞

編製索引之過程簡單說明如下。首先，我們在文稿內挑出欲編入索引之名詞，以 `\index` 加上標識。其次，文稿內須引用排版索引之巨集套件。cW^AT_EX 有一例子檔案 `thesis2.ctx`，置於 `c:\texmf\cwtex\examples\thesis` 檔案夾內，檔案中加入了幾個 `\index` 指令。以下將以此檔案為例，說明排版索引的方法。

以 `cwtex` 及 `latex` 排版之後，文稿內所有以 `\index` 標識之名詞會儲存於另一檔案內。接下來須執行：

```
c:\xtemp>cwidx thesis2
```

對索引名詞排序。之後再執行 `latex` 兩次，即可排出索引。以上之 `cwidx` 是一批次檔，其內含有 cW^AT_EX 所提供之 `cwmkidx` 程式。若使用 WinEdt 文字編輯軟體，其視窗上方的 `cwbibtex` 圖像之下有 `cwidx` 之圖像，其形狀為 **a-z**。點選此一圖像即執行 `cwidx`。排版索引還有一些細節，18.2 節將詳細說明。

根據以上之說明，編排索引的第一步驟是在文稿內以 `\index` 指令標識索引名詞。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}`。此一指令並不影響正文之編排，其功能只是標識此一名詞應列入在索引中，讓 L^AT_EX 可以找出頁碼。

輸入 `\index` 指令時，與名詞本身愈靠近愈好。如果名詞出現於段落中間，`\index` 指令下在段落末端，而段落末端之前恰好換頁，即可能出現下列狀況：名詞出現在第 20 頁，索引卻標識於第 21 頁。此外，下指令時須小心不要讓版面多了不該有的空白。舉例來說，本段文字開頭，我們下了兩個 `\index` 指令：

```
輸入 \verb+\index+ 指令時，
\index{索引!特殊符號}
\index{index@\verb+\index+}
與名詞本身愈靠近愈好。
```

其中，每一個 `\index` 指令各占一行。排版後，第一行末端之“指令時，”之後可能會多出一點空白，原因是兩行 `\index` 指令末端之換行動作各代表一個空白。因此，1-3 行之間合計累積了三個空格。正確的輸入方法如下例所示，`\index` 指令末端加上註銷指令：

```
輸入 \verb+\index+ 指令時，
\index{索引!特殊符號}%
\index{index@\verb+\index+}%
與名詞本身愈靠近愈好。
```

以上的輸入方法讓 1-3 行之空格變成一個。另一種輸入的方法是把原第 3 行指令移至第 2 行末端，再把此行移入第一行末端，而且其中不留任何空格。

回到前面的例子。`\index` 指令會讓「顯示器」一詞單獨成為索引中之一項。如果要把此一名詞排列在「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, 某些索引之頁碼須排為粗體字或上顏色。例如, 某一名詞在書籍內出現多次, 但其中某一頁特別重要, 則標識名詞時可加上排版指令:

```
\index{顯示器|textbf}
```

請注意, 以上之 `textbf` 不須加上反斜線。若此索引項之頁碼為 n , 則 `\index` 指令內之 `|textbf` 等於是讓頁碼以下列方式排版: `\textbf{n}`。亦即, 頁碼將以粗黑體排版。同理, 若 `textbf` 改為 `textit`, 頁碼即以斜體字排版。

利用以上之原理, 我們可以在全文設定區定義以下指令:

```
\usepackage{color}
\newcommand{\idc}[1]{\textcolor{blue}{#1}}
```

其中設定以藍色排版, 則輸入下列之指令:

```
\index{顯示器|idc}
```

將使索引名詞之頁碼以藍色排版。

18.1.1 特殊符號

本書之索引中含有 `\beta` 一詞, 排版指令為 `\verb+beta+`。輸入 `\index` 指令時, 若鍵入 `\index{\verb+\beta+}`, 索引排序時, 因為大括號內是以 `\verb` 起頭, 故此名詞將排列於 `\verb` 處。但正確的位置應該是和以英文字母 `b` 起頭之單字放在一起。為解決此一問題, 輸入時我們須在索引名詞之前加入一排序用文詞, 真正的索引名詞置於其後, 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, 將以 `@` 字元之前的 `beta` 排序。

有些名詞會出現在連續的幾頁或十幾頁中, 欲標識此類名詞, 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, 文稿內連續出現的名詞為 `dvips`, 指令 `|{` (標示開端。結束之處則加入下列指令:

```
\index{dvips@\textsf{dvips}}|)}
```

若開端是在第 12 頁, 結尾是在第 15 頁, 排版結果如下:

dvips, 12–15

由此可知, `\index` 指令內有三個符號, `!`, `@`, 與 `|`, 具有特別之用途。若索引名詞本身含有這三個符號, 必須以其他方式輸入。具體言之, 索引名詞內若含有 `!`, `@`, 與 `|` 三個符號, 輸入在 `\index{...}` 指令內時, 其前面必須加上 `"` 符號, 變成 `"!`, `"@`, 與 `"|`。

18.2 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, 確定書稿中那些名詞或文字要編入索引中, 並以 `\index` 指令標識出來。標識方法, 請見上一節之說明。
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。文稿末端欲排版索引處須加上 `\printindex` 指令。若不上此行指令, \TeX 仍然會進行索引編輯的工作, 但不會將索引列印出來。因此, 排版索引之指令如下:

```
\usepackage{makeidx}
\makeindex
...
\begin{document}
...
\printindex
\end{document}
```

3. 若文稿檔名為 `thesis2.ctx`, 以 `cwtext` 轉換中文, 並以 `latex` 排版之後, 硬碟 `c:\xtemp` 檔案夾內會出現輔助檔案 `thesis2.idx`。此檔案內含索引項目及頁碼, 但尚未排序。每一索引項都置於 `\indexentry` 指令之後; 其內之中文文字為 \TeX 指令格式, 例如:

```
\indexentry{{\MhQ\cH65}\z{\MiQ\cH131}}{7}
...
\indexentry{common goods}{44}
```

首先,第3行之索引為英文名詞 common goods,頁碼為44。第1行之索引為中文名詞,但字碼已轉為 T_EX 字元指令,頁碼是7。若轉回普通中文字,這一行指令為:

```
\indexentry{甘蔗}{7}
```

- 4a. 若為純英文檔案,請進入 DOS,再進入 c:\xtemp 檔案夾內,鍵入以下指令即可將各索引項排序:

```
c:\xtemp>makeindex thesis2
```

排序之後,硬碟內產生 thesis2.ind 檔案。接下來,再執行 latex 兩次,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,請使用 cwidx.bat 批次檔案,執行指令如下:

```
c:\xtemp>cwidx thesis2
```

請注意,勿鍵入附加檔名,否則結果不正確。執行以上指令後,硬碟內將產生 thesis2.ind,最後,再執行 latex 兩次,文稿末端即得索引。

以上是編排索引的一般過程,實際排版時有一些細節可能須特別處理,特別是中文索引部分。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:

```
tex2xtc %1.idx  
cwmkidx %1.xtc  
copy %1.ind cwtemp.ind  
cwtex -- --+ cwtemp.ind  
copy cwtemp.tex %1.ind
```

主要功能是產生中文排序之索引檔案。為了讓中文索引正確排序,第1行指令先執行 tex2xtc 將 thesis2.idx 內以指令形式出現之中文轉回普通中文字,檔名自動取為 thesis2.xtc。第2行指令執行 cwmkidx 程式,將檔案內之中英文索引項排序。排序後之檔名為 thesis2.ind 與 cwtemp.ind。前者之中文字已轉為 T_EX 碼,後者維持普通中文字。

cwmkidx 程式之功能與 T_EX 內附之 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 makeindex 在處理索引時,會將某幾個字元視為



索引排序

指令；而這幾個字元是某些中文字之內碼。因此，若某個中文字之內碼恰好是這幾個特殊字元之一，排序時即會出現問題。

`cxTEX` 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，因此排序結果可能不會百分之百正確，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。

除了調整中文排序之外，我們可能也希望改變索引之編排格式。`LATEX` 內定之索引版面為兩欄格式。如果要改變格式，譬如字體要小一些，或者要以三欄格式排版，最直接的辦法是以 `cxtemp.ind` 檔案為基礎，加入適當之排版指令，再引入文稿之內。底下以 `thesis2.ctx` 為例，說明調整方法。

依以上所述執行排版步驟之後，`c:\xtemp` 檔案夾內會出現 `cxtemp.ind` 檔案，內容如下：

```
\begin{theindex}
  \item common goods, 44

  \indexspace

  \par\bigskip \noindent{\ctxff 3劃} \par\medskip
  \item 三年輪作, 7, \textit{14}, 34

  \indexspace

  \par\bigskip \noindent{\ctxff 5劃} \par\medskip
  \item 甘蔗, 7, 15
    \subitem 三年輪作, 36

  \indexspace

  \par\bigskip \noindent{\ctxff 6劃} \par\medskip
  \item 米糖相剋, 17
\end{theindex}
```

以上每一 `\item` 指令之後即為一索引名詞。欲將以上之索引移入文稿內直接排版，其內容與指令須略作修改為如圖 18.1 所示。本例以 `\chapter*` 指令排版標題，故頁眉之標題以及目錄之標題分別以 `\fancyhead` 及 `\addcontentsline` 指令自自行加入。

```

\newpage
\addcontentsline{toc}{chapter}{索引}
\fancyhead[RO,RE]{索引}
\chapter*{索引}
\xitem common goods, 44

\par\bigskip \noindent{\ctxff 3劃} \par\medskip
\xitem 三年輪作, 7, \textit{14}, 34

\par\bigskip \noindent{\ctxff 5劃} \par\medskip
\xitem 甘蔗, 7, 15
  \subitem 三年輪作, 36

\par\bigskip \noindent{\ctxff 6劃} \par\medskip
\xitem 米糖相剋, 17

```

圖 18.1: 編排索引

若依 \LaTeX 原來的的方法, 整個索引是以 `theindex` 指令環境排版, 其中以 `\item` 指令排版每一索引項目。此一指令之功能與條列指令環境內之同名指令並不相同。因此, 當我們自行排版索引時, 須改用其他指令。圖 18.1 中, 我們將 `\item` 改為 `\xitem`。同時, 在全文設定區作如下之定義:

```
\newcommand\xitem{\par\noindent\hangindent=8mm}
```

除此之外, `cwtemp.ind` 檔案內每隔數行就有 `\indexspace` 指令, 其功能是在英文的索引項目之間插入一點空白。例如, 以字母 a 開頭之索引應與以字母 b 開頭之索引略為分開。不過, 中文索引是以筆劃區隔, 故圖 18.1 中將 `\indexspace` 指令全數刪除。

圖 18.1 之內容是原來 `cwtemp.ind` 檔案之修改結果。下一步驟是把此修改結果複製於主檔案內。`thesis2.ctx` 檔案最後一行指令為 `\end{document}`, 其上為 `\printindex`。因為現在是自行排版索引, 故把 `\printindex` 指令註銷或刪除, 再把圖 18.1 之內容複製於 `\end{document}` 之前。直接執行 `cwtex` 及 `latex` 程式之後, 即可在文稿末端排出索引。

如果是排版書籍, `cwtemp.ind` 之內容可能長達數十頁。較簡單的方法是把圖 18.1 之內容另存新檔, 如 `bookindex.ctx`, 再以 `\include` 指令引入主檔案內。

18.2.1 排版索引之訊息

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, 並將結果記錄於 `cwindexg.ilg` 檔案內, 其內容如下例所示:

```
Scanning input file ... (8 entries accepted, 0 rejected).
Sorting entries ... (30 comparisons).
Generating output file ... (18 lines written, 0 warnings).
Output written in cwindexg.ind.
Transcript written in cwindexg.ilg.
```

第1行之“Scanning input file”表示正檢查原始索引記錄，末端之“0 rejected”訊息顯示原始索引正確無誤。第3行之“Generating output file”表示輸出排序後之結果，末端之訊息“0 warnings”，表示排序後之索引檔也正確無誤。

如果原先標識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。舉例來說，以文字編輯軟體開啓 thesis2.ctx，將檔案內第一個 \index 指令改為：\index{三年輪作@@}。以 latex 排版，並執行 cwidex 排序之後，再以文字編輯軟體開啓 cwindexg.ilg，內容如下：

```
Scanning input file cwindexg.tmp...
!! Input index error (file = cwindexg.tmp, line = 1):
   -- Extra '@' at position 10 of first argument.
.done (7 entries accepted, 1 rejected).
```

第2-3行之訊息表示，cwindexg.tmp 檔案之第1行多了一個@符號。但我們在硬碟內無法找到此一檔案，因為這是一個暫存檔，已遭刪除。

不過，此暫存檔之原始檔案為 thesis2.xtc，儲存於 c:\xtemp 檔案夾內（請注意，附加檔名為 .xtc）。以文字編輯軟體開啓 thesis2.xtc，檢視其第1行，即可知問題所在。以上訊息最後一行之 1 rejected，表示有一項錯誤。除了錯誤訊息之外，還有一些是警告訊息，訊息內會指出問題是在 input=cwindexg.tmp 的第幾行，以及 output=cwindexg.ind 的第幾行。若主檔案為 test.ctx，則 output 所指的 cwindexg.ind 是 c:\xtemp 檔案夾內的 cwttemp.ind，可由之檢查問題的原因。

若直接使用 makeidx 巨集套件排版，索引之首頁會加上 **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}
```

19 書籍排版

排版出一本高品質的書籍是 \TeX 系統的主要目標, 對排版者而言, 則是一項挑戰。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特別的需求, 例如教科書裡, 每一章可能須排版該章之簡要目錄, 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。在說明例子所使用的指令之前, 我們須強調, 排版系統只是工具而已, 高品質排版的先決條件是設計與美感。排版系統好比照相機, 有人手持 Nikon D2X 相機, 照出來的相片卻比傻瓜相機還不如。真的想要排出一本好書, 請先了解書本排版的基本觀念。欲了解排版, 我們推薦 [Bringhurst \(1996\)](#)。此外, [Wilson \(2005\)](#) 為 memoir 巨集套件所寫的使用說明, 也很值得一讀。

19.1 排版設計

排版書籍或長篇文章, 最重要的是全書體例一致。欲做到此點, 我們應儘可能使用巨集指令或巨集套件。譬如, 章節標題、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。

另外一個重點是, 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。簡單的作法是, 第 1 章之檔案取名為 `chap1.ctx`, 第 2 章取名為 `chap2.ctx`, 餘此類推。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, 或者某一章要刪除, 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{market}` 標籤, 文稿內徵引此章時, 僅須使用「參見

第 $\backslash\ref{market}$ 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，有幾種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 會儘可能讓版面右沿切齊，但還是有些情況實在是無法做到版面切齊。此時，我們必須在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。產生此種狀況的原因有很多。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，本頁行距則拉大到填滿整個版面；行距會顯得太大。欲解決此一狀況，我們可以在本頁最後段落末端加上 $\backslash\newpage$ 或者 $\backslash\clearpage$ 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，欲縮減本頁文字版面高度，使版面行數減少一行，可在本頁文字段落之間加入底下指令：

```
 $\backslash\enlargethispage{-\backslashbaselineskip}$ 
```

反之，要將文字版面高度增加0.5公分。指令為 $\backslash\enlargethispage{0.5cm}$ 。

19.2 書籍排版例子

排版書籍通常使用 book 文稿類別。7.3.2 節 (頁 92) 曾介紹一個以 book 文稿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 $c:\text{\textmf}\text{\cwtex}\text{\examples}\text{\book}$ 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 (正文之前的) 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 \TeX 中，這三部分分別稱為 $\backslash\frontmatter$ 、 $\backslash\mainmatter$ ，及 $\backslash\backmatter$ 。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會自動設定頁碼之字體。譬如，如果在正文之前加入 $\backslash\mainmatter$ 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。 \TeX 本身有標準的版面設計，但我們也

可以利用第 17 章所介紹之巨集套件變更設計。自行設計版面時，首先須注意正文與目錄或索引之版面可能不同。通常，正文各章各有編號，但目錄與參考文獻之標題並無編號，故標題可能採不同的設計。

本例子採用分章編排的方法，主檔名為 `CXbook.ctx`。書內計有兩章，分別取名為 `cx-WhatEcon.ctx` 與 `cx-gdp.ctx`，序文內容則為 `cx-intro.ctx`。主檔案之內容全文列於圖 19.1 以供參考。在主檔案內，`\include` (第 22–24 行) 指令應把全書各章之檔案全部引入，但排版時，利用 `\includeonly` (第 14 行) 指令我們可僅排版目前欲修正之檔案。請特別注意，各子檔案須以 `.ctx` 為附加檔名；而且，附加檔名不可鍵入於指令內。

本例中，主檔案納入所有的子檔案，但實際上，我們可以只排版某幾個子檔案。假設在上一輪排版全文時，已產生了全文的頁碼、圖表目錄、數式編號，並假設目前之排版使用 `\includeonly` 指令只引入 `cx-intro.ctx` 與 `cx-gdp.ctx` 兩個檔案，則執行 `latex` 時，僅 `cx-intro.ctx` 與 `cx-gdp.ctx` 兩章會排版出來，但上一輪排版所計算出來的頁碼與圖表目錄資訊仍會納入文稿內。

目錄可以利用 `\tableofcontents` 指令排版，此一指令置於 `cx-intro.ctx` 第 94 行。`TeX` 自動以 `\chapter*` 指令排版其標題。因為序文之標題不編號，故 `\tableofcontents` 指令上方利用 `\titleformat` 設計標題之格式，這是 `titlesec` 巨集套件之指令。參考文獻與索引也是以同樣方法排版。相對而言，第 1 章與第 2 章的標題須編號，我們在 `cx-macro` 巨集指令內定義出 `\numtitle` 以排版各章之標題。

以下簡單說明圖 19.1 所示之主檔案的內容。

- 第 1 行：引入 `cx-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含有中文字，故檔案以 `.ctx` 為附檔名，請參見 14.1.2 (頁 248) 之說明。排版時，`cwtext` 會將檔案內之中文字轉為 `TeX` 指令，檔名則改為 `cx-macro.tex`。
- 7–12 行：選用章節標題之中文字體。
- 第 4 行及 19–20 行：排版時，我們經常須印出結果校對。為了節省紙張，同時也為呈現書本攤開之版面，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 19.3.3 (頁 320) 之說明。若採此一印法，版面位置須略作調整，這是第 4 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 19–20 行指令的作用。以上是完稿之前的作法。完稿時，請記得第 4 行指令應註

```

1 \input{cx-macro}
2
3 % 2-up printing
4 % \hoffset=1cm \voffset=5cm
5 \hoffset=1.5cm \voffset=1.5cm % 1-up
6
7 \ctxdef{chapter}[\ctxff]{\ctxff}
8 \ctxdef{section}[\ctxfm]{\ctxfbb}
9 \ctxdef{subsection}[\ctxfm]{\ctxfbb}
10 \ctxdef{subsubsection}[\ctxfm]{\ctxfk}
11 \ctxdef{paragraph}[\ctxfm]{\ctxfk}
12 \ctxdef{caption}{\ctxfr}
13
14 \includeonly{cx-intro,cx-WhatEcon,cx-gdp}
15
16 \begin{document}
17 \onecolumn
18
19 % \setcounter{page}{0} %% for 2-up printing
20 % \mbox{}\newpage %% for 2-up
21
22 \include{cx-intro}
23 \include{cx-WhatEcon}
24 \include{cx-gdp}
25
26 \titleformat{chapter}[hang]{\vspace*{-2cm}}%
27 \chfont{\parbox{.12\textwidth}{\hfill%
28 \textcolor{dark}{\thechapter}}}%
29 {.09\textwidth}{EMtitle}
30
31 \addcontentsline{toc}{chapter}{\ctxff 參考文獻}
32 \fancyfoot[LE]{\cdot$\the page$\cdot$%
33 \rule{5mm}{0pt} {\small\ctxff 參考文獻}}
34 \fancyfoot[RO]{\small\ctxff 參考文獻}%
35 \rule{5mm}{0pt}$\cdot$\the page$\cdot$}
36
37 \bibliographystyle{ter}
38 \bibliography{cx3test}
39
40 %% Solutions are in two-column format
41 %%
42
43 \newpage
44 \titleformat{chapter}[hang]{\vspace*{-2cm}}%
45 \chfont{\parbox{.12\textwidth}{\hfill%
46 \textcolor{dark}{\thechapter}}}%
47 {.09\textwidth}{EMtitle}
48
49 % The following will change Figure 2.3
50 % to Figure 3.
51 %
52 \renewcommand{\thetable}{\arabic{table}}
53 \renewcommand{\thefigure}{\arabic{figure}}
54 \setcounter{figure}{0}\setcounter{table}{0}
55
56 \fancyfoot[LE]{\cdot$\the page$\cdot$%
57 \rule{5mm}{0pt} {\ctxff 習題解答}}
58 \fancyfoot[RO]{\ctxff 習題解答}\rule{5mm}{0pt}%
59 $\cdot$\the page$\cdot$}
60 \addcontentsline{toc}{chapter}{\ctxff 習題解答}
61
62 % using chngpage.sty to enlarge textwidth
63 %
64 \changetext{}{2cm}{-2cm}{}{}
65 \twocolumn
66
67 \chapter*{\LARGE 習題解答}
68
69 \section*{第1章} \input{ans1}
70 \section*{第2章} \input{ans2}
71
72 \newpage
73 \addcontentsline{toc}{chapter}{\ctxff 索引}
74 \fancyfoot[LE]{\cdot$\the page$\cdot$%
75 \rule{5mm}{0pt} {\ctxff 索引}}
76 \fancyfoot[RO]{\ctxff 索引}\rule{5mm}{0pt}%
77 $\cdot$\the page$\cdot$}
78 \fontsize{10.95}{16pt}\selectfont
79 \printindex
80
81 \changetext{}{-2cm}{+2cm}{}{}
82 \end{document}

```

圖 19.1: CXbook 主檔案

銷, 改用第 5 行; 而 19–20 行指令也須註銷。

- 17 與 65 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` 排版。
- 26–29 行: 以 `\titleformat` 指令設計參考文獻標題之格式。
- 31 行: 參考文獻是以 `\chapter*` 排版標題, 故不會列入目錄內。欲將之加入目錄內, 須使用 `\addcontentsline` 指令。
- 32–35 行: 設計參考文獻之頁眉與頁足。
- 44–47 行: 書末的習題解答與索引之版面較正文的版面寬 2 公分 (64 行與 81 行), 但標題設計與參考文獻 (第 26–29 行) 相同。若標題設計要因應版面變寬而調整, 這幾行指令須修正。
- 52–53 行: 正文各章之圖表編號會自動加入章計數器, 譬如, 第 2 章之表 3 為「表 2.3」。但習題解答中之圖表不加入章編號, 這兩行指令重新定義圖表之編號方式。
- 54 行: 圖表編號之計數器歸零。
- 56–59 行: 設計習題解答之頁眉與頁足。
- 60 行: 參見第 31 行之說明。
- 64 行: `\changetext` 是 `chngpage` 巨集套件之指令, 將文字版面寬度增加 2 公分。第 81 行指令則回復原版面之寬度。
- 73–77 行: 排版索引版面之頁眉與頁足。
- 79 行: 排版索引。

主檔案第 1 行即引入的 `cx-macro.ctx` 巨集指令檔案, 內含所有的巨集套件與指令。指令第 2 行引用 `charter` 巨集套件, 以 Charter 字體排版英文與數字。Charter 字體比中文稍大, 若求完美, 我們可以更改設定, 讓英文字小一些, 請參考 16.1.1 節 (頁 268) 之說明。其他巨集套件與指令, 本書前面各章大都介紹過, 此處不再重覆。此書的第 1 章檔名為 `cx-WhatEcon.ctx`, 檔案內前面幾行如圖 19.2 所示。第 1 行指令為 `\numtitle` (見圖 19.3), 設定標題之排版。第 2 行將 `chapter` 計數器設為 0, 故第 4 行之 `\chapter` 指令將使本章變為第 1 章。

第 6–7 行指令的目的是在各章標題之下排版出該章的目錄, 這是 `titletoc` 巨集套件之指令, 詳見該套件之說明檔。目錄之下接著排版該章之內容簡介, 我們使用迷你指令環境排版。為簡化指令輸入, 我們在定義了一個巨集指令, 取名為 `\firstpage`, 內容請見 `cx-macro.ctx`。

```

1 \numtitle
2 \setcounter{chapter}{0}
3
4 \chapter[經濟學是什麼?]{經濟學是什麼?} \label{em-WhatEcon}
5 \fontsize{12}{20pt}\selectfont
6 \startcontents
7 \printcontents[1]{1}{\setcounter{tocdepth}{1}}
8 \firstpage{
9 這是一個老掉牙的問題。\\
10 每一個人都有自己喜歡的答案， ...
11 }

```

圖 19.2: 書籍排版

圖 19.2 列出第 1 章 `cx-WhatEcon.ctx` 開頭之指令與文字。各章之標題是以 `\numtitle` 指令排版, 其定義如圖 19.3 所示。首先, 我們說明定義中 `\chkeven` 指令之作用。依 \TeX 之內定值, 一般書籍的每一章都是從奇數頁開始排版, 但本例子是從偶數頁開始。為了從偶數頁開始排版, 我們首先在 `\documentclass` 指令內加入 `openany` 選項。因此, 本例子之主檔案一開始所引入的 `cx-macro.ctx` 巨集指令中, 其第 1 行指令如下:

```
\documentclass[12pt,twoside,openany,a4paper]{book}
```

以上指令只是讓各章可以從奇數頁開始, 也可以從偶數頁開始。為了讓各章都一定從偶數頁開始排版, 我們須確認上一章的最後一頁是奇數頁還是偶數頁。如果是在奇數頁結束, 則本章剛好接上。反之, 如果上一章是在偶數頁結束, 則必須先跳空一頁, 再開始排版本章。圖 19.3 為 `\numtitle` 定義之內容, 第 2 行之 `\chkeven` 指令即在確認頁碼是奇數還是偶數。此巨集指令也是定義於 `cx-macro.ctx` 內, 內容如下:

```

\newcommand{\chkeven}{\clearpage\ifodd\count0 \thispagestyle{empty}%
\mbox{} \clearpage \else \fi}

```

指令一開始先以 `\clearpage` 結束上一頁, 接著以 `\ifodd\count0` 確認本頁是否為奇數頁。若為奇數頁, 則以 `\thispagestyle{empty}` 將頁眉與頁足全部空白。接下來, 再以 `\mbox{} \clearpage` 指令跳空一頁。最後, 圖 19.3 第 3–7 行是以 `\titleformat` 設計標題, 第 8–12 行則是設計頁眉與頁足。

本例子含參考文獻及索引, 故以 `latex` 排版之後, 須執行 `cbwibtex` 產生參考文獻, 再執行 `mkidx` 產生索引。若使用 `WinEdt` 軟體, `mkidx` 圖像位於參考文獻 `cbwibtex` 圖像之下。最後, 還至少須跑 `latex` 兩次, 才能得到正確的頁碼。若

```

1 \newcommand{\numtitle}{
2   \chkeven
3   \titleformat{\chapter}[hang]
4     {\vspace*{-2cm}\chfont}
5     {\hspace*{.2\textwidth}\parbox{.12\textwidth}%
6       {\hfill\textcolor{dark}{\thechapter}}}
7     {.09\textwidth}{\EMtitle}
8   \pagestyle{fancy}
9   \fancyhf{}
10  \fancyfoot[LE]{$\cdots\sim\thepage\sim\cdots$\rule{5mm}{0pt}\leftmark}
11  \fancyfoot[RO]{\rightmark\rule{5mm}{0pt}$\cdots\sim\thepage\sim\cdots$}
12  \renewcommand{\headrulewidth}{0pt}}

```

圖 19.3: 章標題之排版

還有後續的修改,但索引頁碼不受影響,則只須執行 `latex` 即可,不須再跑 `mkidx` 程式。

19.3 完稿小工具

TEX 有許多排版的小工具套件,本節將介紹其中幾套。首先,在完稿送印時,通常在版面四角印上裁切記號 (crop marks),以方便裁切及裝訂。欲排版裁切記號,可使用 `crops` 巨集套件。

此外,在完稿之前,我們常須作把全書某一部列印出來。如果你使用 `dvips` 轉換出 PostScript 檔案,我們可以只選擇列印其中部分頁碼。我們也可以透過 `ps2up` 程式將兩頁的版面印於一張紙面上,呈現書籍攤開來之版面。

19.3.1 裁切記號

書籍完稿時,我們應在每一頁版面四個角落加上裁切記號 (cropmarks),書籍印刷之後,裝訂廠即依此記號裁切紙張。欲畫出裁切記號,我們可使用 `crop` 巨集套件,作者為 Melchior Franz。底下是本書裁切記號之指令:

```
\usepackage[cam]{crop}
```

引用巨集套件時,我們加上 `cam` 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。此外,頁面上方會排版檔名,排版日期,及頁碼。

`crop` 巨集套件是依據文稿內所設定之紙張尺寸與正文方塊尺寸排版裁切記號。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。

19.3.2 dvips 程式

排版結果通常是輸出為 PostScript 或者 PDF 檔案。若是前者，目前使用最廣的輸出程式是 `dvips`，作者為 Tom Rokicki。在 WinEdt 視窗內點選 `dvips` 圖像，或按下功能鍵 [F11]，即啟動 `dvips`。不過，有時候我們僅輸出某幾頁，或者執行程式時要加入選項，此時須在 DOS 視窗內鍵入指令與選項。

`dvips` 程式之指令格式如下：

```
c:\xtemp>dvips -option file.dvi
```

附加檔名 `.dvi` 可省略不加。`-option` 為程式選項，底下為較常用之選項。

- p 列印範圍之首頁。譬如，`-p 3` 即選擇自第 3 頁開始列印。
- l 列印範圍之末頁，譬如，`-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。若不作設定，且文稿原名為 `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，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。

19.3.3 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 寫了一套工具程式 `psutils`，主要功能是重新組合版面，本節簡單介紹此套程式之功能。

psutils 包含好幾個程式，一般人最常用的是 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；以方便作書本裝訂。

- 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* 代入數字，表示要將 *n* 頁併入於一頁中。第 2 選項 -p 選用紙張格式，可填入 a4, letter, b5 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。

在 WinEdt 視窗上方之工作列上，dvips 圖像即為 ps2up 之圖像，其形狀為一白紙，上有 PS 字樣。點選此一圖像即啟動 ps2up 批次檔，內容如下：

```
ren %1.ps %1.psa
psnup -2 -s0.8 -pa4 %1.psa %1.ps
del %1.psa
```

第 1 行指令先把原檔案複製一份，附加檔名為 .psa。第 2 行指令啟動 psnup 程式轉換 .psa，轉換結果儲存為 .ps。第 3 行指令刪除 .psa 檔案。

- 調整版面順序: psbook 程式

若某一篇文章共計 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 1 頁內含原來的第 1 與第 2 頁；第 2 頁上則含原來的第 3 與第 4 頁。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。

20 網路出版

對許多人來說,上網已經是生活中不可或缺的一項活動。因此,愈來愈多的人想要了解如何將資訊發表在網路上。網路出版的概念是在 T_EX 系統發展之後才出現的,原始的 T_EX 系統當然不可能包含網路出版的功能。不過,不少 T_EX 的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。本章的目的是從 L^AT_EX 系統的角度說明如何作網路出版 (web publishing)。有關於 L^AT_EX 與網路出版之關係,詳見 [Goossens and Rahtz \(1999\)](#) 之介紹。

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 L^AT_EX 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。

20.1 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直接編排 PDF 並不容易。幸運的是,L^AT_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 \LaTeX 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 \LaTeX 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,一份30頁的稿件,PDF 檔案可能高達 600K 大小。如果文稿內容不太複雜,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 \LaTeX 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 \LaTeX 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,由瀏覽器 (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 \LaTeX 簡單多了。不過,就網路排版而言,HTML 有一些功能是 \LaTeX 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,網路上也有幾套工具程式,可以將 \LaTeX 排版結果轉換為 HTML。本章將介紹的是 \LaTeX 2HTML 程式,其功能相當強,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,我們可以將排版文稿輸出為其中之一。因此,剩下來的問題是要選擇哪一種格式作網路排版? 這個問題就必須由你自行決定了! 底下首先介紹輸出 PDF 之工具程式。

20.2 輸出 PDF 檔案

將 \LaTeX 排版結果轉換為 PDF 有幾個可行的路徑。不過,輸出的品質有好有壞。使用 PDF 格式的主要考慮是排成品質; 其中一個關鍵是使用之字型規格。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。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,以滑鼠點選網頁上所列

出的 PDF 檔名時，瀏覽器會自動啟動 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。

以下將介紹幾種輸出 PDF 檔案的方法。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，例如超連結 (hypertext links) 或者填表 (forms)，最簡單的方法是在 \LaTeX 文稿中使用 `hyperref` 巨集套件之指令。此巨集套件可以和上述的工具程式配合使用，我們將在稍後簡單介紹。

- Acrobat Distiller

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 \LaTeX 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。

- Ghostscript

除了 Distiller 程式外，Ghostscript 也提供工具程式 `ps2pdf` 可將 PostScript 檔案轉換為 PDF。若利用較新版的 Ghostscript 作轉換，PDF 檔案之品質甚佳。文稿內所引用之 EPS 圖形檔案也能正確轉換出來。在 WinEdt 視窗上方有一 `ps2pdf` 圖像（見圖 3.2，頁 41），文稿排版完成後，先以 `dvips` 程式轉換為 PostScript 檔案；再點選此一圖像，即可啟動 `ps2pdf` 程式，輸出 PDF。

- pdf \LaTeX

`ps2pdf` 程式是將 PostScript 檔案轉換為 PDF，`pdflatex` 程式則更進一步，它可以將 \LaTeX 文稿直接排版為 PDF。`pdf \LaTeX` 是由 Hàn Thê Thành 所創作。如果文稿內引用了外製圖形，這些圖形必須是 PDF、JPEG 或 PNG 格式；`pdflatex` 無法直接處理 EPS 圖形。雖然有這些限制，如果你只是偶而轉換 PDF 檔案，`pdflatex` 程式不失為是一個好的選擇。

- dvipdfmx

`dvipdfmx` 以 Mark A. Wicks 所發展的 `dvipdfm` 為基礎，但重要的是，它所輸出的 PDF 可以搜尋中文字詞。作者為 Jin-Hwan Cho 與 Shunsaku Hirata。

在 WinEdt 視窗上, ps2pdf 圖像的右方即為 dvipdfmx 程式之圖像。以 latex 排版之後, 點選 dvipdfmx 程式圖像, 即可由 .dvi 檔案轉換出 PDF。

以上幾種方法都可以轉換出 PDF 檔案; 那麼我們應使用那一種? 簡單來說, 如果你希望 PDF 檔案越小越好, 應使用 ps2pdf 或 Distiller。如果你希望 PDF 檔案可以搜尋中文, 應使用 dvipdfmx 或 Distiller。如果你有許多的 EPS 圖形, 並希望能使用 psfrag 之指令轉換圖形內之字元, 你應該使用的是 ps2pdf, 或者 Distiller。不過, 以上各程式都仍在繼續發展當中, 因此某程式目前所沒有的功能, 未來的修正版很可能會加入。

20.2.1 hyperref 巨集套件

為方便引述 (cross-referencing) 章節、表格、與數式, \TeX 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; 可以由甲文稿連結乙文稿; 甚至可以由甲文稿連結某一網站, 或者發電子信函給某人。為方便說明, 我們稱以上之功能為超連結 (hypertext links)。

如果希望 PDF 文件具有超連結之功能, 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,colorlinks=true]{hyperref}
```

方括號內之 dvips 選項表示要以 dvips 轉換為 PS, 再輸出為 PDF。若使用 pdf \LaTeX 排版, 則選項應為 pdftex。反之, 若使用 dvipdfmx 轉換, 則選項應為 dvipdfm。colorlinks=true 選項的目是讓連結之標籤以彩色排版, 若不加此選項, 標籤將會加上紅色外框, 效果並不好。

底下以一個簡單的例子說明引述表格之功能。若文稿中某一表格之標題是以 `\caption` 指令排版, 其後以 `\label{gtable}` 加上標籤。假設此一表格為表 3, 排版之後, 數字 3 將以紅色排版。文稿內任何地方以「參見表 $\backslash\ref{gtable}$ 」即可引述此一表格。排版之後, 徵引處之數字 3 也是以紅色排版。在 Reader 預覽畫面內, 以滑鼠點選紅色數字 3, 畫面即跳至表格所在之頁面。因此, 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、數式編號也可以徵引。

以上是利用 \TeX 原有之指令以產生連結的功能, 而連結之數字將以紅色排版。如果希望連結的標籤是文字串, 而不只是一個阿拉伯數字, 我們可以使用 hy-

perref 巨集套件所提供之 `\hyperlink` 與 `\hypertarget` 指令; 前者之功能類似 \TeX 之 `\ref` 指令, 後者的功能類似 `\label`。舉例來說, 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` 為自行在文稿特定地方以 `\hypertarget` 指令所定義之標籤, 譬如:

```
\hypertarget{there}
本研究成果之排版 ...
```

排版之後, 以 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。

以上是同一文稿內之連結, 若進一步希望由文稿某處連結特定網址, 或者開啓網頁上之 PDF 檔案, 或者下載免費供使用之程式, 可使用 `\href` 指令。譬如,

```
\usepackage[pdftex,colorlinks=true]{hyperref}
...
參見\href{http://ccms.ntu.edu.tw/~ntut019}{cwTeX 網頁}。
```

第 1 行指令選用 `pdftex` 選項, 故我們須以 `pdflatex` 排版。排版之後, 上述指令將排版為「參見 cwTeX 網頁」, 其中「cwTeX 網頁」是以紅色排版。以 Reader 預覽檔案, 並以滑鼠點選紅色文字, 即可連結到 \TeX 網頁。

如果上述網址存放一 PDF 文稿, 可使用以下之指令下載檔案:

```
參考\href{http://ccms.ntu.edu.tw/~ntut019/teh.pdf}{台灣經濟史} ...
```

同理, 欲透過 ftp 網址以下載檔案, 指令為:

```
請自\href{ftp://ftp1.sinica.edu.tw/pub1/tex/cwTeX}{中研院}下載 ...
```

以上之指令排版之後, 第一個例子中之「台灣經濟史」將為紅色, 以滑鼠點選, 即啓動 Reader, 並開啓 `teh.pdf` 檔案。點選第個例子中之「中研院」, 即可經由 ftp 下載之檔案。

20.2.2 PDF 書籤

以 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)。bookmarks 功能類似普通文稿的章節目錄。對於長篇文稿而言, bookmarks 有其方便之處。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，不過其中文字卻是以 \TeX 轉換後之字型指令，不可能看得懂。

據我們所知，要在中文 PDF 上製作出正確的書籤，目前 (2005 年) 唯一的辦法是以 Adobe Acrobat 商業軟體編輯轉換出來的 PDF 檔案。如果覺得麻煩，我們也可以設定不要書籤，設定指令為 `bookmarks=false`，如下例所示：

```
\usepackage[pdfTeX,colorlinks=true,bookmarks=false]{hyperref}
```

20.3 \TeX 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 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將 \TeX 文稿轉換為 HTML，不同程式處理數式的方法不盡相同。本節所要介紹的是 \TeX 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由程式自動處理，不勞使用者費心。此一程式是由多人合作完成，原始作者為 Nikos Drakos。

為了處理中文稿，`cwTeX` 提供簡單的指令，讓使用者可以設定文稿內較大的中文標題轉換為描點圖形，內文之中文字則直接輸出，不作轉換。以下將以幾個例子說明轉換的方法。

20.3.1 安裝

\TeX 2HTML 是由 perl 語言寫成，實際執行時須使用 `dvips`、`Ghostscript` 及其他工具程式，因此，安裝步驟較為複雜。我們提供之自動安裝程式 `setup12h.bat`，置於光碟 `\util\latex2html` 檔案夾內。安裝步驟如下：

1. 請確定電腦中之 `Ghostscript`、`MiKTeX` 與 `cwTeX` 已能順利執行。若電腦尚未安裝 `perl`，請先進入光碟 `\util` 檔案夾內，執行 `ActivePerl-561.msi`，安裝 `perl` 程式。
2. 安裝程式會自動安裝 `netpbm` 圖形轉換程式，但安裝之前你必須先自行設定其路徑。以文字編輯軟體開啓 `c:` 根目錄下之 `autoexec.bat`，檔案末端新增一行搜尋路徑之之設定指令：

```
path=c:\netpbm\bin;%path%
```

並重新開機,讓設定生效。

3. 重開機之後,點選光碟內 `\util\latex2html` 檔案夾下之 `setupl2h.bat` 即可進行安裝。請注意,安裝程式假設你使用的是 Ghostscript 8.51 版,而且安裝於 `c:\Program Files\gs\gs8.51` 檔案夾內。若你使用其他版本,或安裝於其他硬碟,請先修改 `setupl2h.bat`, 117-118 行之設定,再開始安裝。
4. 如果你的作業系統是 Win2000 或 Windows XP, 安裝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 20 秒,之後按 [Enter] 鍵,再稍候一會兒,程式會繼續執行,並完成安裝。

20.3.2 測試與更新

欲了解是否正確安裝,請進行測試。如果你使用 WinEdt 5.4 版,視窗上方的工作列上的 \TeX 圖像之下有 TtH 圖像,此即代表 \TeX 2HTML 程式。以 WinEdt 開啓 `c:\texmf\cwtex\examples\html` 檔案夾下之 `test-h1.ctx`,直接點選 TtH 圖像,即作排版轉換。(請注意,不須先執行 `cwtex` 或 `latex`。)轉換之結果會儲存於上述檔案夾下之 `\test-h1` 子檔案夾內。

以瀏覽器(如 Internet Explorer)開啓 `test-h1.html`,若檔案內有圖形出現,即表示安裝成功。萬一安裝不成功,請依 `setupl2h.pdf` 之說明手動安裝。該檔案置於光碟 `\util\latex2html` 檔案夾內。如果你不使用 WinEdt 軟體,仍可依以下說明之方法排版。

如上所述,以上之安裝假設你使用 Ghostscript 8.51 版。若未來 Ghostscript 更新為較新版本,如 8.70 版,則設定須更改。以 WinEdt 開啓 `c:\latex2html\bin` 檔案夾內之 `pstoimg.bat`,第 270 行設定 Ghostscript 之版本,請修改為新版本的號碼。

20.3.3 排版方法

以 \TeX 2HTML 輸出 HTML 檔案須經過幾過步驟,`cwTeX` 提供一批次檔 `l2h.bat` 以簡化操作;但我們也可以由 WinEdt 視窗內直接點選 TtH 圖像。在說明執行步驟之前,底下先解釋 \TeX 2HTML 之原理。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, 因此若文稿內有數學式, 我們可將之轉為圖形。

\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 `netpbm` 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。以上的步驟全部自動處理, 不勞使用者費心。

中文也可以用圖形之方式處理。不過, HTML 檔案內之中文有些是正文文字, 有些是標題。中文標題可能選用較大之字級或以彩色排版, 因此, 標題之中文字可以考慮轉為圖形。相對而言, 正文內之中文字則直接以中文碼呈現為宜。因為 \TeX 2HTML 是把數學式全部以圖形檔呈現, 因此要把標題之中文字以圖形方式呈現很簡單: 僅須把中文輸入在數學模式內即可。譬如, `cwTeX` 之網頁是以 \TeX 2HTML 製作, 首頁有一項標題是「總體經濟學」。在製作網頁時, 排版此標題文字之指令如下:

```
\definecolor{title}{cmyk}{0.1,0.65,0.65,0.35}
...
$\mbox{\color{title}{\LARGE \ctxfmues 總體經濟學}}$
```

我們先定義一彩色 `title`, 再以之排版標題。標題的 5 個中文字是置於數學模式內, 但因為是中文, 故必須放在 `\mbox` 指令環境內。排版時, \TeX 2HTML 會把這 5 個中文字轉換成一個圖形檔。

標題之外, 正文內之中文字應直接移入 HTML 文稿內。 \TeX 2HTML 系統設計是以英文為對象; 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, `cwTeX` 提供一巨集套件 `cwtex.sty` 及一對應之 `perl` 程式 `cwtex.perl`, 安裝時已移入系統內。如果是中文稿, 請在全文設定區加入下列指令:

```
\usepackage{cwtex}
```

在作 HTML 轉換時, \TeX 2HTML 會自動引入 `cwtex.perl` 程式, 讓中文字得以正確處理。

20.3.4 排版測試檔

`cwTeX` 提供 4 個測試檔, 存放於 `c:\texmf\cwtex\examples\html` 子目錄之下, 請自行測試排版這 4 個例子, 以了解 \TeX 2HTML 之運作原理。以 `WinEdt` 開啓測試

檔, 例如 test-h2.ctx, 直接點選 TtH 圖像即進行轉換動作。(你不須先執行 cwtex 與 latex 程式。) 如果你不使用 WinEdt, 也可以在 DOS 視窗內直接下指令排版。以 test-h2.ctx 為例, 請進入 DOS, 再進入 c:\texmf\cwtex\examples\html 檔案夾, 鍵入下列指令:

```
c:\texmf\cwtex\examples\html>l2h test-h2
```

即可轉換出 test-h2.html。如果你修改 test-h2.html 之內容, 再執行一次, DOS 視窗內可能會出現下列訊息:

```
... *.* , Are you sure (Y/N)?
```

第一次執行時, 若文稿內使用數式或圖形, \LaTeX HTML 會輸出一些圖形檔存於硬碟內。再次執行時, 如果數式或圖形並無改變, 可直接使用原來的圖形檔, 此時以上問題應回答 N。

以上之 l2h 是一批次檔, 內容如下:

```
del %1\*.*
cwtex %1
latex %1
latex %1
tex2xtc %1.aux
copy %1.xtc %1.aux
cwtex -i %1
call latex2html -no_footnode -no_navigation -noinfo -html_version 3.2 %1
```

最後一行指令是啓動 \LaTeX HTML 程式作轉換, 前7行之指令只是預備動作。若文稿檔名為 test-h2.ctx, \LaTeX HTML 所產生的檔案將存入 \test-h2 檔案夾內。如果文稿有所修改, 再次轉換之前須把舊檔刪除, 故以上批次檔第1道指令先將檔案夾內之舊檔刪除。

\LaTeX HTML 轉換檔案之前須先執行 cwtex 與 latex (2次)。第5-6行指令的目的是處理圖表之標題, 以下將進一步說明。第7行指令為 cwtex -i, 其作用也將於底下說明。如果文稿之轉換須用到圖形檔 (例如, 大標題可能以圖形檔呈現), \LaTeX HTML 會啓動 dvips 程式將圖形轉換為 PostScript, 再進一步轉為描點圖形。因此, 文稿檔案須置於 MiKTeX 的檔案搜尋路徑內。依內設值, MiKTeX 的檔案搜尋路徑包括 c:\texmf 與 d:\tex 在內。如果文稿檔案置於以上兩個檔案夾下即無問題。

以下依序說明4個測試檔之內容。

- test-h1.ctx

test-h1.ctx 為純英文檔，其內引用一外製圖形，檔名為 dmark.eps。文稿內容是標準的 \TeX 檔案，唯一不同的是，我們在全文設定區引入 html 巨集套件：

```
\usepackage{html}
```

轉換之後，c:\texmf\cwtex\examples\html\test-h1 檔案夾下可發現約 20 幾個 GIF 檔案，這些就是文稿內之數式與外製圖形。主檔案則轉成 test-h1.html 與 index.html。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 \TeX 2HTML 產生兩個檔案的原因。如果你是要設計網頁之首頁， \TeX 檔案可直接取名為 index.ctx。

20.2.1 節 (頁 326) 曾說明如何在 PDF 檔案內加入超連結功能，這是透過 hyperref 巨集套件。在 \TeX 2HTML 中要產生超連結之功能，則須透過 html 巨集套件。以瀏覽器開啓 test-h1，可發現註解是排於文稿末端，而文稿內引述註解以及圖表之標籤是以紅色排出，直接點選，畫面即連結到註解或圖表所在位置。

- test-h2.ctx

test-h2.ctx 內含中文，全文設定區引入 html 及 cwtex 兩個巨集套件。前者的目的是超連結，後者則是處理中文。題目是以 \title 指令排版：

```
\title{\mbox{\huge\texttt{cw}\TeX{}} 排版系統}
```

我們把題目放入數學模式中，故整個題目會轉換成一個圖檔。(請注意，題目文字須置於 \mbox 指令內。) 排版之後，在 \test-h2 子目錄下可找到 img1.gif，即為題目之圖形檔。

後續的修改中，我們可能只是修改內文文字，題目之圖形檔並無改變。為避免每次都要花費時間產生圖形檔，我們可將 img1.gif 改名為 title.img，文稿內排版題目之指令則改為：

```
\title{\htmladding{title.gif}}
```

\htmladding 為 \TeX 2HTML 之指令，指示直接取用 title.gif。

test-h2.ctx 在排版題目之指令 \maketitle 之後，有一指令 \ctxfoff。一般而言，cwtex 在處理中文檔案時，會把中文字轉為 \TeX 之字型指令。但在 HTML 檔案內，除了標題或特定之文字段落外，內文之中文應直接排出，不作轉換。如果

我們在文稿內下 `\ctxfoff` 指令, 而且執行 `cwTeX` 時加上 `-i` 選項, 則 `\ctxfoff` 指令以下之中文字即不作轉換。(參見上面所介紹的 `l2h.bat` 批次檔的第 7 行。)

與 `\ctxfoff` 作用相反的是 `\ctxfon`, 指示中文字須作轉換。利用這兩個指令, 我們可以控制將文稿內某段中文轉換為圖形。舉例來說, 某文稿有 3 節, 而節標題須以圖形排版。在第 1 節標題之後, 我們須下 `\ctxfoff`, 指示以下之文字照原樣排出。到了第 2 節標題之前, 我們須下 `\ctxfon` 指令, 指示以下之標題文字須轉換 `TEX` 字型指令, 以便進一步轉換為圖形。標題之後, 請記得加入 `\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。以上之 `\ctxfoff` 與 `\ctxfon` 指令要產生作用, 執行 `cwTeX` 時須加入選項 `-i`。若不加入選項, 這兩個指令毫無作用。

在 `test-h2.ctx` 至 `test-h4.ctx` 等例子檔案內使用了幾個 `html` 巨集套件所提供之超連結指令, 簡單列舉如下:

```
\htmlref{描邊字型}{outline}
\htmladdnormallink{台大}{ftp://ftp.ntu.edu.tw/cwTeX}下載。
\htmladdnormallink{tmwu}{mailto:example@ntu.edu.tw}}
\htmladdnormallink{有意思的文章}{issue.html}}
\htmladdnormallink{討論區}{http://ccms.ntu.edu.tw/index.html}}
```

這些指令例子說明, 我們可以利用 `\htmladdnormallink` 指令連結到檔案, 網頁, 或傳送電子郵件。第 1 個指令連結到同一檔案內的 `outline` 標籤, 而標籤可直接以 `TEX` 之 `\label` 定義。

- `test-h3.ctx`

第 3 個例子是製作 `cwTEX` 網頁之簡化版本, 內含一張貓的圖畫。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」。

- `test-h4.ctx`

第 4 個例子較為複雜, 原因是文稿內之表格以 `\caption` 指令排版中文標題, 並以 `\label` 指令定義標籤, 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」。為了引述圖表, `TEX` 排版時會產生 `test-h4.aux` 輔助檔案, 其中儲存表格之標題文字, 但中文字已轉換為 `TEX` 之字元指令。`TEX2HTML` 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。因此, 我們必須把圖表標題之 `TEX` 之字元指令轉回中文字, 這是上述 `l2h.bat` 批次檔第 5–6 行指令的功能。其他部分的運作與上面的例子相同。

以上僅簡單說 \LaTeX 2HTML 之功能與排版方法, 欲進一步了解, 請見其說明檔案, 置於 `c:\latex2html\docs` 檔案夾內。

21 造字

中文排版的問題是：中文字永遠不足。一般的中文 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爲「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。本章第 1 節介紹一簡單的造字方法。

在 Windows 作業系統上，有許多字型廠商提供 TrueType 字型，但這些字型無法直接用於 c\WTeX 系統上。如果你希望使用 Windows 系統上的字型檔案，可試將 TrueType 字型檔轉換爲 c\WTeX 格式。 c\WTeX 系統發展小組的翁鴻翎先生費心寫了兩套轉換軟體，這兩套軟體可將大部分的 Windows 字型檔轉換爲 c\WTeX 系統之格式。本章第 2 節說明這兩套軟體的使用方法。

21.1 造新字

c\WTeX 所提供的 23 套字體都是 Type 1 字型，如果你有造字軟體，可以自行造出 Type 1 字型。不過，如果你只需要一兩個新字，一個辦法是造出新字圖形檔，再引入文稿內。如果所造之新字是 EPS 描邊圖形檔，排版效果相當不錯。

如果你使用 Word 軟體，經過適當的設定就可以在 Word 軟體內打出 Big5 字集以外之中文字，其中可能就有你想要造的字。舉例來說，在 Big5 字集中並無「琮」一字，但在 Word 軟體內可以鍵入此字。利用以下說明的方法，你在 Word 軟體內所能鍵入的中文，都可以很容易地轉成圖形，再引入 c\WTeX 文稿內。因此，如果你有 Word 軟體，造新字圖形很簡單，不過，首先你必須設定讓軟體可以打出非常用字。以注音輸入法爲例，請由 Windows 之「控制台」進入

地區及語言選項 > 語言 > 詳細資料 > 中文(繁體)–注音 > 屬性

其中，「只顯示 Big5 字集」選項不可打勾。

其次，造新字圖形需使用 PostScript 印表機驅動程式以及 `makeeps` 批次檔，後者必須有 `perl` 程式才能運作。如果你的電腦尚未裝這兩個軟體，請由 c\WTeX 光

碟之 \util 檔案夾 (或網路下載), 直接執行 ActivePerl-561.exe 以及 winstcht.exe, 即可安裝。

以「琮」為例, 造新字圖形的步驟如下:

- 開啓 Word, 選用較大字級 (如 24 點), 鍵入「琮」字,
- 由“檔案 > 列印”, 將此字列印於檔案內。列印時, 印表請選用 PostScript Generic Printer, 並選擇「列印至檔案」, 按「確定」存檔。列印檔案之附加檔名內設值為 .prn。若主檔名為 chon, 請將 chon.prn 改名為 chon.ps。
- 若檔案儲存於 c:\xtemp, 接下來請進入 DOS, 再進入 c:\xtemp, 執行:

```
c:\xtemp>makekeys chon
```

硬碟內即產生 chon.eps 及 chon.pdf, 此即為新字圖形。

請把這兩個檔案移入排版文稿所在之檔案夾, 或者 MiKTeX 之搜尋路徑內, 例如 c:\localtexmf\tex, 排版時, latex 才能找到新字圖檔。

在文稿內欲引用新字圖形, 請在全文設定區引入 graphicx 巨集套件, 並鍵入下列之巨集指令 \cxnew:

```
\usepackage{graphicx}
\newcommand{\cxnew}[1]{\setbox3=\hbox{中}\raisebox{- .1\ht3}%
  {\includegraphics[width=.96\wd3]{#1}\ignorespaces}}
```

經過以上設定, 文稿內輸入「王\cxnew{chon}\cxnew{chon} (2000) 之研究」, 排版結果為「王琮琮 (2000) 之研究」。E_TEX 之放大字級指令也可以放大新造字。例如, 「{\Large 王\cxnew{chon}美}」, 將排版成:

王琮美

因此, 新造字與原有之字體會同步放大或縮小。

以上是以 Windows 內附之細明體所造之新字圖形。各家字體廠商所提供之字型, 其字體大小不完全相同, 故造字時可能須略作調整。上述 \cxnew 巨集指令的第 2 行, \wd3 代表新造字之原始寬度。因為 Windows 之細明體較 c_WTeX 明體字為寬, 故引入圖形時我們將之縮小為 0.96。此外, 細明體字型之位置較高, 故指令第一行中以 \raisebox 指令將字體位置下降原始高度 \ht3 的 10%。

Word 軟體內, 任何中文字都可以選用斜體, 粗體, 或粗斜體。現如果造出的是斜體字 琮, 以 \cxnew 指令引入文稿後, 會發現新字右端之空白太大。為解決這個問題, 我們可定義另一個巨集指令:

```
\newcommand{\cxnewi}[1]{\setbox3=\hbox{中}\raisebox{-.1\ht3}%  
{\includegraphics[width=\wd3]{#1}\ignorespaces}\kern-.1\wd3}
```

其中，字寬維持原始寬度，而字體右端加入 `\kern-.1\wd3` 指令，指示字距縮減新字寬度的 10%。利用 `\cxnewi` 引入斜體字，排版結果為：

王琼美

21.2 由 TrueType 輸出 cwT_EX 字型檔

如果你有偏好的 TrueType 中文字型檔，利用本節所介紹之工具程式即可將之轉換為 cwT_EX 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，都受到智慧財產權法律之保障。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，但使用此程式所可能涉及的法律問題，概與作者無關。

21.2.1 `cwpt1` 與 `cwttf` 軟體

字型轉換工具有兩套，分別名為 `cwpt1` 與 `cwttf`。這兩套程式集存放於 cwT_EX 光碟的 `\util\TrueType` 檔案夾內。第一套程式集 `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。所謂 CJK 系統是一套可以處理中文 (Chinese)，日文 (Japanese)，與韓文 (Korean) 之 T_EX 延伸系統，故名為 CJK 系統，作者為 Werner Lemberg。翁鴻翎修改並加強 `ttf2pfb` 程式之功能，以便能轉換出 cwT_EX 系統格式之字型檔。此程式可處理傳統的字型檔，也可以轉換較新的 unicode 字型檔。

第二套程式集 `cwpt1` 的源頭是 `ttf2pt1` 程式，作者是 Mark Heath。這是由 Andrew Weeks 的 `ttf2pfa` 修改而來，後續之維護者是 Sergey Babkin。翁鴻翎修改並加強 `ttf2pfb` 程式之功能，使之可以轉換出 cwT_EX 系統格式之字型檔，並且可處理傳統的及 unicode 之字型檔。¹

依測試，`cwpt1` 執行時需較長時間，但品質似較佳。此外，`cwpt1` 程式集也可以轉換簡體字字型，故以下之說明以 `cwpt1` 程式集為主。TrueType 字體與 Type 1 字體之規格不同，一般而言，轉換後之字型品質不如原有的。有時候，某些字甚至可能出現錯誤。另外，某些公司之字體是以字根組合方式造字，如華康公司的某些字體以及 Windows 系統之細明體，這些字體可能無法作轉換。

¹特別感謝李果正先生幫忙測試 Linux 版本。

表 21.1: cwTeX 字體名稱

字體	-l	-	-b	-u	-x
b (黑體)	\ctxfbl	\ctxfb	\ctxfbb	\ctxfbu	\ctxfbx
f (仿宋體)	\ctxffl	\ctxff			
k (楷體)	\ctxfkl	\ctxfk		\ctxfku	\ctfkk
l (隸體)		\ctxfl	\ctxflb		
m (明體)	\ctxfml	\ctxfm	\ctxfmb	\ctxfmu	\ctxfmx
n					
o					
p					
r (圓體)	\ctxfrl	\ctxfr	\ctxfrb	\ctxfriu	\ctxfrix

21.2.2 轉換方法

TrueType 字型檔案有傳統格式與 unicode 兩種格式。取得 TrueType 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 為附加檔名。譬如, 全真中明體字型檔原名為 avmmv.ttf, 超研澤之粗楷體字型檔原名為 wk3.ttf。

由 TrueType 字型檔轉換為 Type 1 字型檔需要好幾個步驟, cwTeX 提供一批次檔 makefont.bat 以簡化動作, 其功能如下:

- 啟動 cwpt1 程式轉換 TrueType 字型為 Type 1 字型,
- 在 Type 1 字型檔內加入 unicode 字碼,
- 產生使用字型所需之檔案, 如 .fd, 並複製於 MiKTeX 系統內。

請先建立一工作檔案夾, 例如 c:\cwfont, 取得 TrueType 字型後, 請將之複製於其內, 改名為 1.ttf。字型轉換軟體置於 cwTeX 光碟 \util\TrueType 檔案夾內, 請將 cwpt1.zip 亦解壓於 c:\cwfont。

執行 makefont.bat 之前, 我們須作幾個設定。首先, 我們須決定轉換後之字型檔名。如果 cwTeX 字體指令取名為 \ctxfn, 則此字型檔應取名為 n。表 21.1 列出 cwTeX 現有之字體, 及可選用之字體名字。表中若列出某字體之指令, 如 \ctxfm 或 \ctxfriu, 表示該字體已經存在。相反的, 「空格」表示可供使用之字體, 但字型檔尚不存在。因此, 我們可選用之字體名稱包括 \ctxfn, \ctxfo 與 \ctxfo 等。如果你要轉換的是粗楷體, 字體指令應為 \ctxfkb。

以文字編輯軟體開啓 makefont.bat, 大約在 24–26 行有 3 行指令如下:

```
set fontname=KB
set sfm=kb
set scale=0.94
```

若字體指令為 `\ctxfn`，請將 KB 改為 N，並 kb 改為 n（請注意大小寫）。上述第3行指令設定字體縮小比率為 0.94，轉換程式即以此設定值縮放字體。

`makefont.bat` 第 30 行指令啟動 `cwpt19.exe`，這是轉換傳統 TrueType 字型檔所用的程式。指令中之 `%sfm%` 參數將代入上述之定義值 kb，`%scale%` 則代入 0.94。接下來之 `-n 0.95` 選項，將字體在水平方向再縮小 0.95，變成長方形。以上之數值都可以重新設定，請見下一小節之說明。

經過以上之設定之後，請進入 DOS，再進入 `c:\cwfont` 檔案夾，確定其中已有 `1.ttf` 字型檔，鍵入以下指令：

```
c:\cwfont>makefont
```

此一批次檔相當複雜，執行時間也較長。若一切無誤，執行之後產生下列檔案：

- `n0.pfb` 至 `n51.pfb` 等 52 個字型檔：置於 `c:\texmf\fonts\type1\cwtext`,
- `n0.tfm` 等 208 個字型檔：置於 `c:\texmf\fonts\tfm\cwtext`,
- `n0.afm` 等 52 個字型檔：置於 `c:\texmf\fonts\afm\cwtext`,
- `ot1cwN0.fd` 等 208 個字型定義檔：置於 `c:\texmf\tex\latex\cwtext`,
- `cwtextn.map` 字型對應檔：置於 `c:\texmf\dvips\cwtext`,
- `updmap.cfg` (置於 `c:\localtexmf\miktex\config`) 會加入 `cwtextn.map` 字型對應檔之設定

字型檔產生之後，以 `\ctxfn` 指令即可選用新字型。例如，`{\ctxfn 測試}` 指令將使兩個中文字以新字型排版。

上述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，則僅會產生 `p0.pfb` 至 `p21.pfb`。

以上所轉換的是 Windows 傳統字型檔。若是 unicode 字型檔，其附加檔名為 `.ttc`。例如，華康 unicode 細明體檔名為 `Dfn_m3.ttc`。轉換時，先將字型檔複製為 `1.ttc`。上述 `makefont` 批次檔第 30 行之指令應改為 `cwpt1u9`。除了 `cwpt1` 程式之外，我們也可使用 `cwttf` 程式集作轉換。欲使用此一程式，應解壓縮 `cwttf.zip`，並執行其中之 `makefont` 批次檔。此一批次檔之第 30 行啟動的是

cwttf.exe, 這適用於傳統 TrueType 字型檔。若為 unicode 字型檔, 則 cwttf.exe 應以 cwttfu.exe 替代。

21.2.3 更改設定

在說明如何更改字型轉換之設定之前, 我們先解釋字型轉換程式各選項之意義。以全真中明體為例, 此字型檔傳統規格, 故應使用 cwpt1 程式集內之 cwpt19 程式。若轉換後之字體指令為 \ctxfn, 則轉換指令如下例所示:

```
c:\cwfont>cwpt19 n -d 0.94 -n 0.95 -l 0.05 -s 0.167 -e 0.91
```

為了解各選項之意義, 我們先對字型設計作一點背景說明。我們所轉換之字型是要用於 \TeX 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。為了與 \TeX 原有之英數字體以及 $\text{cw}\TeX$ 中文字體匹配, 轉換字型檔時, 其大小須細心設定。此外, Windows 之 TrueType 字體通常較為方正, $\text{cw}\TeX$ 系統所提供的字體則略呈長方形。因此, 我們還須將字體調整成長方形。

由以上的背景說明, 以下進一步說明各選項之意義。

- 第 1 選項 n 是新字型名稱。本例之字型取名為 n, 轉換成功之後, 若欲使用此字型, 指令為 \ctxfn。目前, $\text{cw}\TeX$ 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。表 21.1 (頁 338) 列出可使用之字體名稱。
- 第 2 選項為 -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為長方形。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, 其作用是讓轉換後之字型檔的每一個字的位置, 往上提升原中文字高度的 5%。
- 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為 16.7 度, 狹長字之壓縮比率為 91%。

如果更改 -d 與 -n 之值, 則 c:\cwfont 檔案夾內 cwtexm.map 之設定值也須對應修改。此一檔案內有如下之設定:

```
m0 CWTEX-M0 ".95 ExtendFont" <m0.pfb
...
me0 CWTEX-M0 ".8645 ExtendFont" <m0.pfb
```

```
...
mes0 CWTEX-M0 ".8645 ExtendFont .167 SlantFont" <m0.pfb
...
```

以上例子之第一行設定標準字體,最後一行設定狹長斜體,中間一行則為狹長字體之設定。其中, $.8645 = 0.95 \times 0.94$ 。如果 `cwpt19` 之設定更改如下: 等比例縮小數值為 `-d=0.93`, 水平再縮小比率為 `-n=0.92`, 則 `cwtextm.map` 內之 `.95` 應改為 `.92`, 而 `.8645` 等改為 `.8556 (= 0.93 \times 0.92)`。

21.3 專業造字軟體

`cwTeX` 系統使用的是 Type 1 中文字。如果你有專業造字軟體,如 `Fontographer`, 可由第 1 節之圖形字進一步造出 Type 1 字型檔。`cwTeX` 的第 52 個字型檔 (`m52.pfb` 與 `bb52.pfb`) 存放三十幾個新造字,而一個 Type 1 字型檔可存放 256 個字,因此新造字可加入 `m52.pfb` 字型檔內。相關細節,請見造字軟體之使用手冊。

把新字圖形加入 Type 1 字型檔之後,造字軟體可以輸出新的 Type 1 以及相關之字型檔案;但我們僅須留存 `m52.pfb` 與 `m52.afm` 兩檔案,其餘可刪除。接下來,我們須使用 `afm2tfm` 程式由 `m52.afm` 產生 `m52.tfm`。若字型檔案置於 `c:\xtemp` 檔案夾內,請進入 DOS,執行:

```
c:\xtemp>afm2tfm m52
```

最後,請將 `TeX` 字型度量檔 `m52.tfm` 移入 `c:\texmf\fonts\tfm\cwtext` 檔案夾內, Type 1 字型檔 `m52.pfb` 移入 `c:\texmf\fonts\type1\cwtext` 檔案夾;而 PostScript 字型度量檔 `m52.afm` 則移入 `c:\texmf\fonts\afm\cwtext` 檔案夾內。

21.3.1 使用新字型

若我們造的是圖形字,則利用 `\cxnew` 或 `\cxnewi` 即可使用新字。如果我們造的是 Type 1 字型檔,使用上反而較麻煩。以「琮」為例,雖然 `Word` 軟體上可顯示此字,但一般的文字編輯軟體,如 `WinEdt`,則沒辦法。

`cwTeX` 所用的間接辦法如下: 從次常用字中選用一字代表新造字。譬如,我們可選用「椋」代表「琮」。假設新造字「琮」加入於 `m52` 字型檔第 98 個字碼位置,則以上之資訊應記錄於 `c:\texmf\cwtext` 之 `cwfont.usr` 檔案內,方式如下:

```
椋 52 98 % m
```

第1欄為新造字之代替字,第2欄為字型檔編號,第3欄為新造字之編號。cwtex程式轉換中文時,遇有次常用字,如「棕」時,會先到 cwfont.usr 檔案搜尋是否有字元對應。若有對應,該次常用字即選用 m52 字型檔之第98個字排版。如此,文稿內鍵入替代字「棕」,但排版出來的是「琮」。如果我們同時造明體字與粗黑體字,則記錄行之末端可標示為 % m, bb。

21.3.2 中文字之排序

日常生活中常用之中文字大約在三千到四千之間。Windows 系統之中文字型含常用字與次常用字,共計 13,053 字,其中,常用字 5,401 字。若加入非常用字,字數更多。cwTeX 大部分之字型檔都含 13,053 字,但少數字型檔僅含 5,401 字。

cwTeX 中文字型亦有常用字與次常用字之區分,但編碼順序與 Big5 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,大部分字型檔都含 256 字,次序從 0 排至 255。但 m0 字型檔內之中文字則從第 44 排至第 255, m21 字型檔內從 0 排至 238。

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 位置存放 65 個次常用字;第 87–255 位置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。

最後,21.3.1 節(頁 341)說明如何以間接方法使用新造字。若不嫌麻煩,新造字也可以加入 Windows 系統內,並指定一內碼。此一作法的好處是,我們可以在 Windows 之應用軟體內鍵入新造字。爲了讓 cwTeX 能夠處理新造字,指定之內碼必須在下列範圍內: FA40-FEFE, 8E40-A0FE, 及 8140-8DFE。另外,新造 Type 1 字型檔之編號必須是在 52–57 之間,如 m52.pfb 或者 bb54.pfb。

22 排版訊息

排版過程難免出現錯誤。錯誤的原因很多, 包括:

- 忘了輸入右大括號,
- 鍵入左大括號時誤為左圓括號,
- 指令鍵入錯誤, 例如, `\footnote` 誤為 `\fotnote`,
- 指令的用法錯誤。

程式執行時, `cwtext` 與 `latex` 會嘗試判斷錯誤的原因。但是, 有些錯誤實在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, `latex` 會認為左大括號以下的文字全部都是註解。連續幾頁的文字都排入註解之後, 程式中容納註解文字之空間不足, 乃發出 `TeX capacity exceeded ...` 之訊息。從字面上來看, 似乎是 `TeX` 之記憶體太小, 但真正的原因是忘了輸入右大括號。

解讀錯誤與警告訊息 (warning) 的能力愈強, 我們就能愈快改正錯誤, 完成排版工作。底下, 22.1 節說明排版程式的一般訊息。22.2 節說明 `cwTeX` 之錯誤與警告訊息, 22.3 節則說明 `ETEX` 之訊息。

22.1 排版訊息

執行各排版程式時, 顯示器上會出現一些訊息。這些訊息也會記錄在訊息檔案內。第一類訊息的內容是記錄處理過程及結果, 第二類訊息則指出錯誤所在。以 3.2 節 (頁 39) 之 `test.ctx` 為例, 執行 `cwtext` 程式時, 顯示器上將出現下列訊息:

```
This is cwTeX preprocessor, version 18.3g.  
**test.ctx  
(test.ctx [7])  
Output written on c:\xtemp\test.tex & cinput.tex.
```

第 1 行末端的 18.3 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。以上的訊息內容也記錄於 `c:\xtemp` 之 `test.xlg` 檔案內。因此，如果來不及從視窗內讀取訊息，我們仍可事後閱覽此一檔案以了解排版之過程。

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會出現排版訊息；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 e-TeX, Version 3.141592-2.2 (MiKTeX 2.4) (preloaded format ...
entering extended mode
**test.tex
(test.tex
...
Output written on test.dvi (1 page, 644 bytes).
```

以上訊息的第 3 行表示 `latex` 處理的是 `test.tex` 檔案，第 4 行之後的 ... 為數十行關於使用字型之訊息。最後一行

```
Output written on TEST.dvi (1 page, 644 bytes).
```

說明版面計有為 1 頁，排版結果儲存於 `test.dvi` 檔案中，大小為 644 bytes。

如果你使用 `pdflatex` 排版，訊息內容與 `latex` 類似。一般而言，`latex` 與 `pdflatex` 之訊息較為完整。如果你從 `cwtext` 之訊息無法找到錯誤之所在，應進一步檢視 `latex` 之訊息。`cwtext`，`latex`，以及各工具程式之訊息檔案的附加檔名如下：

- `cwtext`: `.xlg`
- `latex` 或 `pdflatex`: `.log`
- `cwbibtex` (參考文獻): `.blg`
- `cwidx` (索引): `.ilg`

以上之訊息檔案皆儲存於 `c:\xtemp` 檔案夾內。排版長篇文稿或書籍時，應檢視每一個訊息檔案，以確保沒有錯誤。

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，其後的文字都不再處理。`latex` 所記錄的錯誤訊息都會指出錯誤可能是發生在那一行，如果由行數還是無法確認錯誤之原因，我



偵測錯誤

們可以在該行之後的兩三行處下 `\end{document}` 指令，重新排版，看看錯誤訊息是否仍與原來的相同。`\end{document}` 指令上移之後，排版文稿已縮短，較容易找出問題所在。

若仍無法找到問題所在，我們可以在此行之上數十行處再下 `\end{document}` 指令。排版之後，如果錯誤訊息已消失，表示錯誤是發生於兩個 `\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。以下簡單說明 `cwTeX` 與 `latex` 之訊息。

22.2 `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` 時，若文稿內使用了中文字型檔以外之中文字，`cwTeX` 會發出訊息，並說明錯誤發生於檔案的那一行，版面上將該字所在位置將留為空白。

舉例來說，有人習慣使用「1 ~ 5」說明數字之範圍。在文字編輯軟體內，我們可直接鍵入「~」符號，但 `cwTeX` 程式會對此發出錯誤訊息：

```
! Line 5: X is not character of cffont. Ignore.
```

其中 `X` 為鍵入之中文字或符號。錯誤的原因是，`cwTeX` 中文字型檔內並無此一符號。如果你要排版以上符號，應使用 `ETEX` 之指令：`$$sim$`。

除此之外，`cwTeX` 程式會檢查文稿檔案內可能的格式錯誤。首先，輸入指令時左右大括號必須對稱出現。若文稿內的右大括號數目少於左大括號，轉換程式即發出下列訊息：

```
! Too many {'s.
```

反之，若文稿的最後一行的 `\end{document}` 指令遺漏了左大括號，而使文稿之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，或者暫時無法確定錯誤的原因，可按 [Enter] 鍵繼續轉換。反之，若鍵入 `x`，即中斷程式。

`ETEX` 文稿是以 `\end{document}` 結尾，因此，`cwTeX` 程式會檢查檔案結尾處是否有此指令。輸入時若遺漏反斜線，最後一行變成 `end{document}`，`cwTeX` 會發出下列錯誤訊息：

```
! I can't find \end{document}.
```

latex 排版時須使用字型度量檔，附加檔名為 .tfm。在轉換中文時，cwtex 會同時檢查硬碟內是否有所需之字型度量檔，若找不到，即發出底下訊息：

```
! I can't find xx.tfm
```

其中，xx.tfm 為所缺之字型度量檔。譬如，若所列檔名為 b2.tfm，表示硬碟中並無中黑字型檔。

22.3 latex 或 pdflatex 訊息

latex (或 pdflatex) 所產生之訊息，部分來自 \TeX ，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 3.2 節 (頁 39) 之 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。轉換中文時，cwtex 會發出如上一節所述之警告訊息，但按下 [Enter] 之後仍可將全文轉換完畢。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 ! 符號之後的訊息表示錯誤的原因，或 \TeX 所自動採取的補救方法。此例中訊息內容為 Missing } inserted.，表示 \TeX 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} 與 {\MeS\cH171}。這些符號是中文字經過轉換後之 \TeX 字型指令。

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 程式將繼續執行,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。除了 [Enter] 之外,我們也可以鍵入以下字母,其效果如下:

- S latex 將繼續編排文稿至最後一頁,中間遇有錯誤,原則上並不停下來。所有之訊息會存於 test.log 檔案中,
- 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,此一選項則不停。
- Q 與按 R 鍵類似。但 R 鍵使排版訊息出現於顯示器上,此一選項則使訊息只儲存於 test.log 中,不出現在顯示器上。
- H 求助選項, latex 將試著分析錯誤的原因,並提出解決之道。
- X 立即中斷程式,停止編排。

如果你安裝的 $\text{cwt}_{\text{E}}\text{X}$ 是從網路下載,其中含5套字體,包括明體,粗黑,圓體,楷體,以及仿宋體。如果文稿內使用了其他字體,排版之後,中文字將變成亂碼。或者,你本來是要下 $\backslash\text{ctxfm}$ 指令,但不小心指令變成 $\backslash\text{ctxfn}$ 。排版之後,中文字也是亂碼。latex 之 .log 檔案內將出現下列訊息:

```
LaTeX Font Warning: Some font shapes were not available, defaults  
substituted.
```

意思是說, latex 找不到字型檔,故自動以其他字型替代。

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 $\backslash\text{footnote}$ 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 $\$\sqr{\beta}\$,$ 將出現下列錯誤訊息:

```
! Undefined control sequence.  
<recently read> \sqr
```

```
1.6 $\sqr  
      {\beta}$.  
?
```

latex 不會說鍵入的指令錯誤, 只會提醒說 `\sqr` 指令未曾定義過。

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。可能的原因是指令環境之名字輸入錯誤, 或者忘了引入巨集套件。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, 排版軟體即發出下列訊息:

```
! LaTeX Error: Can be used only in preamble.
```

本章前言中說明 “TeX capacity exceeded” 之訊息。從字面上來看, 此項訊息是說 TeX 程式之記憶體空間不足, 無法處理複雜的文稿。但是, 如本章一開頭的例子所示, 真正原因是因為 `\footnote{...}` 指令中忘了加上右大括號。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, 若連續有十幾個圖表放在一起, 可能會出現此項錯誤訊息。如果大部分圖表都是占一整頁空間, 解決的辦法是直接以指令控制圖表出現之位置, 不要使用浮動圖表指令環境。

執行 latex 時須鍵入文稿檔名。若是長篇文稿, 我們可能以一主檔案引入各章檔案。若文稿檔名鍵入錯誤, 或者置於某檔案夾內但 latex 無法找到, latex 即發出底下訊息:

```
! I can't find file '...'  
Please type another input file name =
```

解決的辦法是輸入正確檔名。萬一硬碟中暫時無任何 `.tex` 檔案可以輸入, 可試著輸入 `null`, 這代表一虛擬檔案, latex 應會繼續排版動作。

23 cwTeX 系統

cwTeX 系統包含工具程式與字型檔兩大部分。 cwTeX 系統將文稿內之中文轉換為 $\text{T}_{\text{E}}\text{X}$ 字型指令,再交由 $\text{T}_{\text{E}}\text{X}$ 或 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 排版。目前, cwTeX 系統主要配合 Windows 作業系統上之 $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$, 以及 Linux 作業系統之 $\text{T}_{\text{E}}\text{X}$ 系統。但本版之 Linux 版本並未更新。

cwTeX 系統可分中文轉換程式與工具程式, 以及字型及相關工具程式兩部分, 以下簡單介紹。

23.1 中文轉換程式與字型檔

cwTeX 系統之中文中文轉換程式及工具程式主要有下列:

- **cwtext**

主要功能是將文稿內之中文轉換為 $\text{T}_{\text{E}}\text{X}$ 字型指令, 以進一步由 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 排版。若文稿檔案名為 `test.ctx`, 轉換後之檔案取名為 `test.tex`。在 DOS 視窗內直接鍵入 `cwtext`, 按 [Enter], 即顯示所有之選項。

- **tex2xtc**

將 `.tex` 內之 $\text{T}_{\text{E}}\text{X}$ 字型指令轉回 Big5 中文, 此程式所作之轉換為 `cwtext` 之反向, 其功能之一是救回原文稿檔案。如果你不小心把 `.ctx` 刪掉, 但 `c:\xtemp` 檔案夾內之 `.tex` 仍留存, 可利用此程式救回。若原檔名為 `test.ctx`, `cwtext` 程式將之轉換為 `test.tex`, 鍵入以下指令:

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

硬碟內將產生 `test.xtc`。請注意, 附加檔名為 `.xtc`。除了部分中文字體指令外, 此檔案之內容與 `test.ctx` 相同。

除了救回檔案之外, 此程式之主要用途如下。排版書籍時, `latex` 會在硬碟中產生一些輔助檔案, 以編排目錄、參考文獻、及索引, 附加檔名分別為 `.toc`,



救回原稿

.bb1, 及 .ind。以上輔助檔案內之中文已轉為 \TeX 字型指令。以目錄為例, \TeX 會使用內定之格式, 如果要自行設計目錄之版面, 可使用 `tex2xtc` 將 .toc 檔案內之中文字型指令轉為 Big5 中文, 再進一步編輯排版。

- `cwibibtex`

延伸 $\text{Bi}\TeX$ 之功能, 以處理中文參考文獻。

- `cwmkidx`

延伸 `makeindex` 之功能, 以處理中文索引。

$\text{cw}\TeX$ 系統之中文字型檔如下:

- 23套橫排中文 Type 1 字型

$\text{cw}\TeX$ 提供23套中文 Type 1 字型。若使用2005年的新版字型檔, 排版結果以 Adobe Distiller 或 `dvipdfmx` 輸出 PDF, 檔案可搜尋中文字詞。

- 5套直排中文 Type 1 字型

- 5套中文 TrueType 字型

$\text{cw}\TeX$ 提供5套中文 TrueType 字型, 存放於光碟 `\util\TrueType` 檔案夾內。這些字型可用於 Windows 系統上的軟體, 如 Word 或者 Corel Draw。

此外, 我們可使用 `cwpt1` 與 `cwttf` 工具程式將 Windows 系統上之 TrueType 字型檔轉換為 $\text{cw}\TeX$ 字型檔。

以上工具程式及字型檔係使用於繁體中文系統。除此之外, $\text{cw}\TeX$ 尚有簡體字版本。有意使用者, 請進入 $\text{cw}\TeX$ 之 ftp 網址, 再進入 `\cwTeX-GB` 子目錄, 參閱其中之說明檔。

以上之軟體適用於 Windows 作業系統。 $\text{cw}\TeX$ 尚有 Linux 作業系統之版本, 不過, 此一系統之程式大都是2002年之版本, 並未隨著更新。有意使用者, 請進入 $\text{cw}\TeX$ 之 ftp 網址, 再進入 `\cwTeX_Linux` 子檔案夾, 參閱其中之說明檔。

23.2 舊式字體指令

舊式中文字體指令是指 `\m12` 與 `\f11` 等指令格式, 前者選用明體 12 點字體, 後者選用仿宋 11 點字體。如果你的文稿所參考的其他人使用之老舊檔案, 其中可能使用舊式字體指令。舊式指令同時選用字體與字級點數, 新式指令, 如 `\ctxfm` 或 `\ctxff` 則選用字體, 字級則由 \TeX 之指令決定。新指令的優點是, 中英文字級大小較為一致。

請注意,只有 \LaTeX 才能使用新式字體指令, \TeX 文稿不能用。若文稿內新舊兩種指令夾雜使用,排版時可能出現錯誤。常見的問題是,中文字大小不一。如果你的 \LaTeX 文稿檔案內仍使用舊式字體指令,我們強烈建議更新為新指令,方法如下:以文字編輯軟體將文稿內所有的 $\backslash m10, \backslash m11, \backslash m12$ 等指令替代為 $\backslash ctxfm$ 。同理,將所有的 $\backslash bb10, \backslash bb11, \backslash bb12$ 等指令替代為 $\backslash ctxfbb$ 。

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。若要求使用新式字體指令,選項為 $-s$:

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

若是要使用舊式字體指令,選項為 $-f$ 。在 DOS 之下,直接鍵入以上指令即可執行。我們也可以將選項加入文字編輯軟體之設定內,以 WinEdt 為例,請由

```
Options > Menu Setup > &Accessories
```

勾選 Menu Items 下之 cwtex , 中間欄之 Macro 內有如下之設定

```
Run("cwtex.exe -c -d=c:\xtemp %N%T", "%P");
```

其中已有 $-c$ 與 $-d$ 兩個選項。 $-s$ 選項應加在 cwtex.exe 之後, $\%N\%T$ 之前。三個選項之順序並不重要,可任意排列。

請注意,如果你不是使用 \LaTeX 排版,而是使用 \TeX , 因為 \TeX 無法辨識新式字體指令,故你只能使用舊式中文字體指令。

23.2.1 調整中文字距

cwtex 程式有許多選項,有些人抱怨 cwtex 內定之中文字距太小。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,使得排版文稿難以閱讀。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,若文稿檔名為 test.ctx , 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
```

請注意, `-10.5` 中第 2 個符號為英文字母 `l`, 不是阿拉伯數字 `1`。如果是要上移中文字, 選項數字應為負值, 如 `-1-0.3`。

此外, 排版中文大字標題時, 有時候須微調某兩個字之間距。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 `\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。

參考書目

- Bringhurst, Robert (1996), *The Elements of Typographic Style*, Vancouver: Hartley & Marks, 2nd edition.
- Goossens, Michel and Rahtz, Sebastian (1999), *ET_EX Web Companion*, Reading, Massachusetts: Addison Wesley.
- Goossens, Michel, Rahtz, Sebastian, and Mittelbach, Frank (1997), *ET_EX Graphics Companion*, Reading, Massachusetts: Addison Wesley.
- Grätzer, George (2000), *Math into ET_EX*, Boston: Birkhäuser.
- Knuth, Donald E. (1990), *The T_EXBook*, Reading, Massachusetts: Addison Wesley.
- (1999), *Digital Typography*, Leland, California: Stanford Junior University.
- Kopka, Helmut and Daly, Patrick W. (1995), *A Guide to ET_EX2e*, Reading, Massachusetts: Addison Wesley, 2nd edition.
- Mittelbach, Frank and Goossens, Michel (2004), *The LaTeX Companion*, Reading, Massachusetts: Addison Wesley, 2nd edition.
- Pakin, Scott (2005), “The comprehensive ET_EX symbol list”, URL: [On CTAN at: info/symbols/comprehensive](#).
- Reckdahl, Keith (1997), “Using imported graphics in ET_EX2 ϵ ”, URL: [On CTAN at: info/epslatex](#).
- Wilson, Peter (2005), “The memoir class for configurable typesetting user guide”, URL: [On CTAN at: texmf/tex/latex/memoir](#).

數學符號索引

\backslash acute (\acute{o}), 146 \backslash aleph (\aleph), 145 \backslash alpha (α), 136 \backslash amalg (\amalg), 144 \backslash angle (\angle), 145 \backslash approx (\approx), 141 \backslash approxeq (\cong), 142 \backslash arccos, 137 \backslash arcsin, 137 \backslash arctan, 137 \backslash arg, 137 \backslash ast ($*$), 144 \backslash asymp (\asymp), 141 \backslash backepsilon (ϵ), 142 \backslash backprime (\backprime), 145 \backslash backsim (\sim), 142 \backslash backsimeq (\simeq), 142 \backslash backslash (\backslash), 145, 150 \backslash bar (\bar{o}), 146 \backslash barwedge ($\bar{\wedge}$), 144 \backslash Bbbk (\mathbb{k}), 145 \backslash because (\because), 142 \backslash beta (β), 39, 136 \backslash beth (\beth), 145 \backslash between ($\text{\textcircled{}}\text{\textcircled{}}$), 142 \backslash bigcap (\bigcap), 138 \backslash bigcirc (\bigcirc), 144 \backslash bigcup (\bigcup), 138	\backslash bigodot (\bigodot), 138 \backslash bigoplus (\bigoplus), 138 \backslash bigotimes (\bigotimes), 138 \backslash bigsqcup (\bigsqcup), 138 \backslash bigstar (\bigstar), 145 \backslash bigtriangledown (\bigtriangledown), 144 \backslash bigtriangleup (\bigtriangleup), 144 \backslash biguplus (\biguplus), 138 \backslash bigvee (\bigvee), 138 \backslash bigwedge (\bigwedge), 138 \backslash blacklozenge (\blacklozenge), 145 \backslash blacksquare (\blacksquare), 145 \backslash blacktriangle (\blacktriangle), 145 \backslash blacktriangledown (\blacktriangledown), 145 \backslash blacktriangleleft (\blacktriangleleft), 142 \backslash blacktriangleright (\blacktriangleright), 142 \backslash bot (\perp), 145 \backslash bowtie (\bowtie), 141 \backslash Box (\square), 145, 186 \backslash boxdot (\boxdot), 144 \backslash boxminus (\boxminus), 144 \backslash boxplus (\boxplus), 144 \backslash boxtimes (\boxtimes), 144 \backslash breve (\breve{o}), 146 \backslash bullet (\bullet), 144 \backslash bumpeq (\bumpeq), 142 \backslash Bumpeq (\Bumpeq), 142 \backslash cap (\cap), 144	\backslash Cap (\cap), 144 \backslash cdot (\cdot), 144 \backslash cdots (\cdots), 148 \backslash centerdot (\centerdot), 144 \backslash check (\check{o}), 146 \backslash chi (χ), 136 \backslash circ (\circ), 144 \backslash circceq ($\circ\equiv$), 142 \backslash circlearrowleft (\circlearrowleft), 141 \backslash circlearrowright (\circlearrowright), 141 \backslash circledast (\circledast), 144 \backslash circledcirc (\circledcirc), 144 \backslash circleddash (\circledash), 144 \backslash circledS (\circledS), 145 \backslash clubsuit (\clubsuit), 145 \backslash complement (\complement), 145 \backslash cong (\cong), 141 \backslash coprod (\coprod), 138 \backslash cos, 137 \backslash cosh, 137 \backslash cot, 137 \backslash coth, 137 \backslash csc, 137 \backslash cup (\cup), 144 \backslash Cup (\cup), 144 \backslash curlyeqprec (\curlyeqprec), 142 \backslash curlyeqsucc (\curlyeqsucc), 142 \backslash curlyvee (\curlyvee), 144
---	--	--

$\backslash curlywedge$ (\curvearrowright), 144
 $\backslash curvearrowleft$ (\curvearrowleft), 141
 $\backslash curvearrowright$ (\curvearrowright), 141
 $\backslash dagger$ (\dagger), 144
 $\backslash daleth$ (\beth), 145
 $\backslash dashv$ (\dashv), 141
 $\backslash ddagger$ (\ddagger), 144
 $\backslash ddot$ ($\ddot{}$), 146
 $\backslash ddots$ (\cdots), 148
 $\backslash deg$, 137
 $\backslash delta$ (δ), 136
 $\backslash Delta$ (Δ), 136
 $\backslash det$, 137
 $\backslash diagdown$ (\searrow), 145
 $\backslash diagup$ (\swarrow), 145
 $\backslash diamond$ (\diamond), 144
 $\backslash Diamond$ (\Diamond), 145
 $\backslash diamondsuit$ (\diamondsuit), 145
 $\backslash digamma$ (Γ), 145
 $\backslash dim$, 137
 $\backslash div$ (\div), 144
 $\backslash divideontimes$ (\oslash), 144
 $\backslash dot$ ($\dot{}$), 146
 $\backslash doteq$ (\doteq), 141
 $\backslash Doteq$ (\Doteq), 142
 $\backslash doteqdot$ (\doteqdot), 142
 $\backslash dotplus$ ($\dot{+}$), 144
 $\backslash doublebarwedge$ ($\overline{\wedge}$), 144
 $\backslash doublecap$ ($\overline{\cap}$), 144
 $\backslash doublecup$ ($\overline{\cup}$), 144
 $\backslash downarrow$ (\Downarrow), 140, 150
 $\backslash Downarrow$ (\Downarrow), 140, 150
 $\backslash downdownarrows$ (\Downarrow), 141
 $\backslash downharpoonleft$ (\harpoonleft), 141
 $\backslash downharpoonright$ (\harpoonright), 141
 $\backslash ell$ (ℓ), 145
 $\backslash emptyset$ (\emptyset), 145
 $\backslash epsilon$ (ϵ), 136
 $\backslash eqcirc$ (\equiv), 142
 $\backslash eqsim$ (\simeq), 142
 $\backslash eqslantgtr$ (\gtrsim), 142
 $\backslash eqslantless$ (\lesssim), 142
 $\backslash equiv$ (\equiv), 141
 $\backslash eta$ (η), 136
 $\backslash eth$ (\eth), 145
 $\backslash exists$ (\exists), 145
 $\backslash exp$, 137
 $\backslash fallingdotseq$ (\fallingdotseq), 142
 $\backslash Finv$ (\Finv), 145
 $\backslash flat$ (\flat), 145
 $\backslash forall$ (\forall), 145
 $\backslash frac$, 134
 $\backslash frown$ (\frown), 141
 $\backslash Game$ (\Game), 145
 $\backslash gamma$ (γ), 136
 $\backslash Gamma$ (Γ), 136
 $\backslash gcd$, 137
 $\backslash ge$ (\geq), 140
 $\backslash geq$ (\geq), 141
 $\backslash geqq$ (\geqq), 142
 $\backslash geqslant$ (\gtrsim), 142
 $\backslash gets$ (\leftarrow), 139
 $\backslash gg$ (\gg), 141
 $\backslash ggg$ (\ggg), 142
 $\backslash gggtr$ (\gggtr), 142
 $\backslash gimel$ (\beth), 145
 $\backslash gnapprox$ (\gtrapprox), 143
 $\backslash gneq$ (\gtrsim), 143
 $\backslash gneqq$ (\gtrsim), 143
 $\backslash gnsim$ (\gtrsim), 143
 $\backslash grave$ ($\grave{}$), 146
 $\backslash gtrapprox$ (\gtrapprox), 142
 $\backslash gtrdot$ (\gtrdot), 142
 $\backslash gtreqless$ (\gtrless), 142
 $\backslash gtreqqless$ (\gtrless), 142
 $\backslash gtrless$ (\gtrsim), 142
 $\backslash gtrsim$ (\gtrsim), 142
 $\backslash gvertneqq$ (\gtrneqq), 143
 $\backslash hat$ ($\hat{}$), 146
 $\backslash hbar$ (\hbar), 145
 $\backslash heartsuit$ (\heartsuit), 145
 $\backslash hom$, 137
 $\backslash hookleftarrow$ (\hookleftarrow), 140
 $\backslash hookrightarrow$ (\hookrightarrow), 140
 $\backslash hslash$ (\hbar), 145
 $\backslash Im$ (\Im), 145
 $\backslash imath$ (\imath), 145, 146
 $\backslash in$ (\in), 141
 $\backslash inf$, 137
 $\backslash infinity$ (∞), 145
 $\backslash int$ (\int), 138
 $\backslash intercal$ (\intercal), 144
 $\backslash iota$ (ι), 136
 $\backslash jmath$ (\jmath), 145, 146
 $\backslash Join$ (\Join), 141
 $\backslash kappa$ (κ), 136
 $\backslash ker$, 137
 $\backslash lambda$ (λ), 136
 $\backslash Lambda$ (Λ), 136
 $\backslash langle$ (\langle), 150
 $\backslash lceil$ (\lceil), 150
 $\backslash ldots$ (\dots), 148
 $\backslash le$ (\leq), 140
 $\backslash leadsto$ (\leadsto), 140
 $\backslash left.$, 151
 $\backslash leftarrow$ (\leftarrow), 140
 $\backslash Leftarrow$ (\Lleftarrow), 140
 $\backslash leftarrowtail$ (\leftarrowtail), 141
 $\backslash leftharpoondown$ (\leftharpoondown), 140
 $\backslash leftharpoonup$ (\leftharpoonup), 140
 $\backslash leftleftarrows$ (\Lleftarrow), 141
 $\backslash leftrightarrow$ (\leftrightarrow), 140
 $\backslash Leftrightarrow$ (\Lrightarrow), 140
 $\backslash leftrightarrows$ (\Lleftrightarrow), 141

\leftrightharpoons (\Leftrightarrow), 141
 \leftrightsquigarrow (\Leftrightarrow), 141
 \leftthreetimes (λ), 144
 \leq (\leq), 141
 \leqq (\leq), 142
 \leqslant (\leq), 142
 \lessapprox (\lesssim), 142
 \lessdot (\lessdot), 142
 \lesseqgtr (\lesseqgtr), 142
 \lesseqgtr (\lesseqgtr), 142
 \lessgtr (\lesseqgtr), 142
 \lesssim (\lesssim), 142
 \lfloor (\lfloor), 150
 \lg , 137
 \lhd (\triangleleft), 144
 \lim , 137
 \liminf , 137
 \limsup , 137
 \ll (\ll), 141
 \llcorner (\llcorner), 145
 \Lleftarrow (\Leftarrow), 141
 \lll (\lll), 142
 \llless (\llless), 142
 \ln , 137
 \lnapprox (\lnapprox), 143
 \lneq (\lneq), 143
 \lneqq (\lneqq), 143
 \lnsim (\lnsim), 143
 \log , 137
 \longleftarrow (\longleftarrow), 140
 \Longleftarrow (\Longleftarrow), 140
 \longleftrightarrow (\longleftrightarrow), 140
 \Longleftrightarrow (\Longleftrightarrow), 140
 \longmapsto (\longmapsto), 140
 \longrightarrow (\longrightarrow), 140
 \Longrightarrow (\Longrightarrow), 140
 \looparrowleft (\looparrowleft), 141
 \looparrowright (\looparrowright), 141
 \lozenge (\diamond), 145
 \lrcorner (\lrcorner), 145
 \Lsh (\Lsh), 141
 \ltimes (\ltimes), 144
 \lvertneqq (\lvertneqq), 143
 \mapsto (\mapsto), 140
 \max , 137
 \measuredangle (\measuredangle), 145
 \mho (\mho), 145
 \mid (\mid), 141
 \min , 137
 \models (\models), 141
 \mp (\mp), 144
 μ (μ), 136
 \multimap (\multimap), 141
 ∇ (∇), 145
 \natural (\natural), 145
 \ncong (\ncong), 143
 \nearrow (\nearrow), 140
 \neg (\neg), 145
 \neq (\neq), 141
 \nexists (\nexists), 145
 \ngeq (\ngeq), 143
 \ngeqq (\ngeqq), 143
 \ngeqslant (\ngeqslant), 143
 \ngtr (\ngtr), 143
 \ni (\ni), 141
 \nleftarrow (\nleftarrow), 141
 \Nleftarrow (\Nleftarrow), 141
 \nleftrightarrow (\nleftrightarrow), 141
 \Nleftrightarrow (\Nleftrightarrow), 141
 \nleq (\nleq), 143
 \nleqq (\nleqq), 143
 \nleqslant (\nleqslant), 143
 \nless (\nless), 143
 \nmid (\nmid), 143
 \notin (\notin), 140
 \notin (\notin), 140
 \nparallel (\nparallel), 143
 \nprec (\nprec), 143
 \npreceq (\npreceq), 143
 \nrightarrow (\nrightarrow), 141
 \Nrightarrow (\Nrightarrow), 141
 \nshortmid (\nshortmid), 143
 \nshortparallel (\nshortparallel), 143
 \nsim (\nsim), 143
 \nsubseteq (\nsubseteq), 143
 \nsubseteqq (\nsubseteqq), 143
 \nsucc (\nsucc), 143
 \nsucceq (\nsucceq), 143
 \nsupseteq (\nsupseteq), 143
 \nsupseteqq (\nsupseteqq), 143
 \ntriangleleft (\ntriangleleft), 143
 \ntrianglelefteq (\ntrianglelefteq), 143
 \ntriangleright (\ntriangleright), 143
 \ntrianglerighteq (\ntrianglerighteq), 143
 ν (ν), 136
 \nvdash (\nvdash), 143
 \nwarrow (\nwarrow), 140
 o (o), 136
 \odot (\odot), 144
 \oint (\oint), 138
 ω (ω), 136
 Ω (Ω), 136
 \ominus (\ominus), 144
 \oplus (\oplus), 144
 \oslash (\oslash), 144
 \otimes (\otimes), 144
 $\overbrace{a+b}$, 146
 $\overleftarrow{}$, 140

<code>\overline</code> ($\bar{}$), 146	<code>\rightleftarrows</code> (\Leftrightarrow), 141	<code>\subseteq</code> (\subseteq), 142
<code>\overrightarrow</code> , 140	<code>\rightleftharpoons</code> (\rightleftharpoons), 140, 141	<code>\subsetneq</code> (\subsetneq), 143
<code>\parallel</code> (\parallel), 141	<code>\rightrightarrows</code> (\Rrightarrow), 141	<code>\subsetneqq</code> (\subsetneqq), 143
<code>\partial</code> (∂), 145	<code>\rightsquigarrow</code> (\rightsquigarrow), 141	<code>\succ</code> ($>$), 141
<code>\perp</code> (\perp), 141	<code>\rightthreetimes</code> (\llcorner), 144	<code>\succapprox</code> (\gtrsim), 142
<code>\phi</code> (ϕ), 136	<code>\risingdotseq</code> ($\dot{=}$), 142	<code>\succcurlyeq</code> (\succcurlyeq), 142
<code>\Phi</code> (Φ), 136	<code>\Rrightarrow</code> (\Rrightarrow), 141	<code>\succeq</code> (\geq), 141
<code>\pi</code> (π), 136	<code>\Rsh</code> (\Rsh), 141	<code>\succnapprox</code> (\gtrsim), 143
<code>\Pi</code> (Π), 136	<code>\rtimes</code> (\rtimes), 144	<code>\succneqq</code> (\succneqq), 143
<code>\pitchfork</code> (\pitchfork), 142	<code>\searrow</code> (\searrow), 140	<code>\succnsim</code> (\succnsim), 143
<code>\pm</code> (\pm), 144	<code>\sec</code> , 137	<code>\succsim</code> (\succsim), 142
<code>\Pr</code> , 137	<code>\setminus</code> (\setminus), 144	<code>\sum</code> (\sum), 138
<code>\prec</code> ($<$), 141	<code>\sharp</code> (\sharp), 145	<code>\sup</code> , 137
<code>\precapprox</code> (\lesssim), 142	<code>\shortmid</code> (\shortmid), 142	<code>\supset</code> (\supset), 141
<code>\preccurlyeq</code> (\preccurlyeq), 142	<code>\shortparallel</code> (\shortparallel), 142	<code>\Supset</code> (\Supset), 142
<code>\preceq</code> (\leq), 141	<code>\sigma</code> (σ), 136	<code>\supseteq</code> (\supseteq), 141
<code>\precnapprox</code> (\lesssim), 143	<code>\Sigma</code> (Σ), 136	<code>\supseteqeq</code> (\supseteqeq), 142
<code>\precneqq</code> (\precneqq), 143	<code>\sim</code> (\sim), 141, 345	<code>\supsetneq</code> (\supsetneq), 143
<code>\precnsim</code> (\precnsim), 143	<code>\simeq</code> (\simeq), 141	<code>\supsetneqq</code> (\supsetneqq), 143
<code>\precsim</code> (\precsim), 142	<code>\sin</code> , 137	<code>\surd</code> (\surd), 145
<code>\prime</code> ($'$), 145	<code>\sinh</code> , 137	<code>\swarrow</code> (\swarrow), 140
<code>\prod</code> (\prod), 138	<code>\smallfrown</code> (\smallfrown), 142	<code>\tan</code> , 137
<code>\propto</code> (\propto), 141	<code>\smallsetminus</code> (\smallsetminus), 144	<code>\tanh</code> , 137
<code>\psi</code> (ψ), 136	<code>\smallsmile</code> (\smallsmile), 142	<code>\tau</code> (τ), 136
<code>\Psi</code> (Ψ), 136	<code>\smile</code> (\smile), 141	<code>\therefore</code> (\therefore), 142
<code>\rangle</code> (\rangle), 150	<code>\spadesuit</code> (\spadesuit), 145	<code>\theta</code> (θ), 136
<code>\lceil</code> (\lceil), 150	<code>\sphericalangle</code> (\sphericalangle), 145	<code>\Theta</code> (Θ), 136
<code>\Re</code> (\Re), 145	<code>\sqcap</code> (\sqcap), 144	<code>\thickapprox</code> (\approx), 142
<code>\restriction</code> (\restriction), 141	<code>\sqcup</code> (\sqcup), 144	<code>\thicksim</code> (\thicksim), 142
<code>\rfloor</code> (\rfloor), 150	<code>\sqsubset</code> (\sqsubset), 141, 142	<code>\tilde</code> ($\tilde{}$), 146
<code>\rhd</code> (\rhd), 144	<code>\sqsupset</code> (\sqsupset), 141, 142	<code>\times</code> (\times), 144
<code>\rho</code> (ρ), 136	<code>\sqsupseteq</code> (\sqsupseteq), 141	<code>\to</code> (\rightarrow), 140
<code>\right..</code> , 150	<code>\square</code> (\square), 145	<code>\top</code> (\top), 145
<code>\rightarrow</code> (\rightarrow), 140	<code>\star</code> (\star), 144	<code>\triangle</code> (\triangle), 145
<code>\Rrightarrow</code> (\Rrightarrow), 140	<code>\subset</code> (\subset), 141	<code>\triangledown</code> (\triangledown), 145
<code>\rightarrowtail</code> (\rightarrowtail), 141	<code>\Subset</code> (\Subset), 142	<code>\triangleleft</code> (\triangleleft), 144
<code>\rightharpoondown</code> (\rightharpoondown), 140	<code>\subseteq</code> (\subseteq), 141	<code>\trianglelefteq</code> (\trianglelefteq), 142
<code>\rightharpoonup</code> (\rightharpoonup), 140	<code>\subteq</code> (\subseteq), 141	<code>\trianglelefteq</code> (\trianglelefteq), 142

<code>\triangleright</code> (\triangleright), 144	<code>\upuparrows</code> (\Uparrow), 141	<code>\vartriangleright</code> (\triangleright), 142
<code>\trianglerighteq</code> (\trianglerighteq), 142	<code>\urcorner</code> (\urcorner), 145	<code>\vdash</code> (\vdash), 141
<code>\twoheadleftarrow</code> (\twoheadleftarrow), 141	<code>\varepsilon</code> (ε), 136	<code>\VDash</code> (\Vdash), 142
<code>\twoheadrightarrow</code> (\twoheadrightarrow), 141	<code>\varkappa</code> (\varkappa), 145	<code>\Vdash</code> (\Vdash), 142
<code>\ulcorner</code> (\ulcorner), 145	<code>\varnothing</code> (\emptyset), 145	<code>\vdot</code> ($\dot{\cdot}$), 148
<code>\underbrace</code> ($\underbrace{a+b}$), 146	<code>\varphi</code> (φ), 136	<code>\vec</code> (\vec{d}), 146
<code>\unlhd</code> (\unlhd), 144	<code>\varpi</code> (ϖ), 136	<code>\vee</code> (\vee), 144
<code>\unrhd</code> (\unrhd), 144	<code>\varpropto</code> (\propto), 142	<code>\veebar</code> (\veebar), 144
<code>\uparrow</code> (\uparrow), 140, 150	<code>\varrho</code> (ϱ), 136	<code>\Vvdash</code> (\Vvdash), 142
<code>\Uparrow</code> (\Uparrow), 140, 150	<code>\varsigma</code> (ς), 136	<code>\wedge</code> (\wedge), 144
<code>\updownarrow</code> (\updownarrow), 140, 150	<code>\varsubsetneq</code> (\varsubsetneq), 143	<code>\widehat</code> (\widehat{d}), 146
<code>\Updownarrow</code> (\Updownarrow), 140, 150	<code>\varsubsetneqq</code> (\varsubsetneqq), 143	<code>\widetilde</code> (\widetilde{d}), 146
<code>\upharpoonleft</code> (\upharpoonleft), 141	<code>\varsupsetneq</code> (\varsupsetneq), 143	<code>\wp</code> (\wp), 145
<code>\upharpoonright</code> (\upharpoonright), 141	<code>\varsupsetneqq</code> (\varsupsetneqq), 143	<code>\wr</code> (\wr), 144
<code>\uplus</code> (\uplus), 144	<code>\vartheta</code> (ϑ), 136	<code>\xi</code> (ξ), 136
<code>\upsilon</code> (υ), 136	<code>\vartriangle</code> (\triangle), 145	<code>\Xi</code> (Ξ), 136
<code>\Upsilon</code> (Υ), 136	<code>\vartriangleleft</code> (\triangleleft), 142	<code>\zeta</code> (ζ), 136

索引

英文索引在前, 中文索引從 369 頁開始。粗體字之頁碼代表指令之定義, 或重要例子。「常見問題」請見 372 頁。數學符號索引請見 355 頁。

- !‘ (¡), 57
- % (註銷指令), 57, 128, 239, 284
- & (分隔欄位), 149, 163
- ’ (’) (英文右單引號), 63
- ’’ (”) (英文右雙引號), 63
- (, 150
- \((進入隨文數學模式), 133
- \[(進入展式數學模式), 133
- \$ (隨文數學模式), 133
- \$\$ (展式數學模式), 133
-), 150
- \) (離開隨文數學模式), 133
- \“{o} (ò 重音符號), 56
- \. {o} (ó 重音符號), 56
- /, 150
- ?‘ (¿), 57
- [, 150
- \ (TeX 指令開頭), 58
- \ (空格指令), 60
- \# (#), 58
- \\$ (\$), 58
- \% (%), 58
- \& (&), 58
- \, (加入小空白), 63
- \, (加大數學符號間距), 159
- \: (加大數學符號間距), 159
- \; (加大數學符號間距), 159
- | (界限符號), 150
- \\ (換行), 110
- * (換行), 110
- ˆ (數式上標), 58
- _ (數式下標), 58
- \{ (左大括號), 134, 150
- |, 150
- \} (右大括號), 134, 150
- \] (離開展式數學模式), 133
-], 150
- \~{o} (ô 重音符號), 56
- \‘{o} (ò 重音符號), 56
- ‘ (’) (英文左單引號), 63
- ‘‘ (”) (英文左雙引號), 63
- \~{o} (ô 重音符號), 56
- ~ (加入空白), 58, 194
- \! (縮小數學符號間距), 159
- \aa (å), 57
- \AA (Å), 57
- \abovecaptionskip, 177
- abstract 指令環境, 96
- \abstractname, 96, 252
- accents (重音符號), 56
- Acrobat Distiller, 325
- Acrobat Reader, 37, 324
- \addcontentsline, 91, 97, 106, 295
- \address, 234
- \addtocontents, 106
- \addtocounter, 251
- \addtolength, 82, 172
- Adobe Acrobat, 328
- Adobe Illustrator, 204, 275
- \ae (æ), 57
- \AE (Æ), 57
- afm2tfm, 274, 320
- \alert, 229
- align 指令環境, 156
- align* 指令環境, 156, 158
- \alph, 99, 125
- \Alph, 99, 125
- \alsiname, 252
- $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$, 131
- $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$, 152
- amsmath 巨集套件, 137, 152, 155, 158
- amssymb 巨集套件, 131, 140, 141
- amsthm 巨集套件, 152, 155
- \and, 95

answers 巨集套件, 241, 243
 \appendix, 107, 108
 \appendixname, 108, 252
 \appendixpage, 108
 appenix 巨集套件, 108
 \arabic, 99, 252
 array 指令環境, 148, 162
 array 巨集套件, 87, 162, 163, 169, 172
 \arraycolsep, 170
 \arrayrulecolor, 185
 \arrayrulewidth, 170
 \arraystretch, 170
 Arseneau, Donald, 58, 169, 220
 article 文稿類別, 39, 84, 86, 89, 299
 \atop, 147
 \author, 23, 89, 94, 229
 autoexec.bat, 50
 avant 巨集套件, 268

 \b{o} (ø 重音符號), 56
 babel 巨集套件, 56, 65, 266
 Babkin, Sergey, 337
 \backmatter, 104, 314
 Barratt, Craig, 202
 Barroca, Leonor, 188, 239
 \baselineskip, 82
 \baselinestretch, 76
 beamer 巨集套件, 35, 227, 232
 \belowcaptionskip, 177
 Berry, Karl, 276, 278
 Bezos, Javier, 99, 283, 296
 \bf, 69, 70
 \bfseries, 69
 bibentry 巨集套件, 264
 \bibhang, 264
 \bibliography, 254, 256, 264
 bibliography style (文獻資料格式), 254
 \bibliographystyle, 254, 264
 \bibname, 252, 254
 \bibsep, 264
 bibtex, 20, 253
 BiBT_EX, 253
 \big, 139, 149
 \Big, 139, 149
 \bigg, 149
 \Bigg, 149
 \bigskip, 29, 112
 \bline, 184
 bm 巨集套件, 137
 \bm, 137
 bmeps 巨集套件, 208
 \bmod, 145
 \boldmath, 137
 \boldsymbol, 137
 book 文稿類別, 86, 92, 289, 299, 314
 bookman 巨集套件, 268
 bookmarks (PDF 書籤), 230, 327
 booktabs 巨集套件, 162, 163, 166, 167
 \bottomrule, 166, 191
 bounding box, 199, 209
 bmeps, 209
 ebb, 210
 \bpara, 183
 Braams, Johannes, 65
 Bringhurst, Robert, 65, 313
 \btable, 250

 \c{o} (ø 重音符號), 56
 caption 巨集套件, 177, 190
 \caption, 106, 175, 176, 178, 199, 211, 252, 333
 \captionsetup, 177
 Carlisle, David, 78, 173, 185, 188, 192, 202, 224, 249
 \cc, 234
 ccfonts 巨集套件, 269
 \ccname, 252
 \cdot, 304
 center 指令環境, 111, 164, 199
 \centering, 110, 164, 287
 chancery 巨集套件, 268
 \changetext, 286, 317
 \chapter, 96, 286, 301
 \chapter*, 294
 \chaptermark, 300
 \chaptername, 99, 252
 charter 巨集套件, 268
 chngpage 巨集套件, 286, 317
 Cho, Jin-Hwan, 325
 \choose, 147
 Christie, Agatha, 18
 \circle, 222, 223
 \citeauthor, 263
 \citep, 262
 \citet, 262
 CJK 系統, 337
 \cleardoublepage, 105
 \clearpage, 105, 240, 314
 \cline, 166, 170
 \closing, 233
 cm (長度單位), 82
 \cmidrule, 166
 CMYK, 218
 color 巨集套件, 184, 187, 216, 218, 224, 289
 \color, 178, 217, 229, 230
 color model, 218
 CMYK, 218
 RGB, 218

Duggar, Angus, 320
 dvipdfm, 210
 dvipdfmx, 41, 198, 199, 210, 213, 325, 326, 350
 dvips, 274, 279, 320, 325

 ebb, 210
 Eijkhout, Victor, 128
 em (長度單位), 82
 \em, 70
 em-dash (---), 64
 emTeX, 15, 276
 en-dash (--), 64
 Encapsulated PostScript (EPS), 199
 \encl, 234
 \enclname, 252
 \endfirsthead, 191
 \endfoot, 191
 \endhead, 191
 \endlastfoot, 191
 endnotes 巨集套件, 125
 \enlargethispage, 106, 314
 enumerate 指令環境, 115, 117, 229, 250
 environment (指令環境), 73
 EPS (Encapsulated PostScript), 199, 201, 205
 makeeps, 207
 PostScript 印表機驅動程式, 204
 描點圖形, 208
 輸出, 204
 epstopdf, 210
 equation 指令環境, 133, 156
 \evensidemargin, 85
 ex (長度單位), 82
 Excel, 201, 204, 205, 207
 \extrarowheight, 164, 170

 \fancyfoot, 303
 fancyhdr 巨集套件, 87, 93, 102, 106, 283, 298, 304
 \fancyhead, 303
 \fancypagestyle, 304
 fancyvrb 巨集套件, 128, 130
 \fbox, 121
 \fboxrule, 122
 \fboxsep, 122, 130, 187, 217, 218
 Fear, Simon, 161, 166
 figure 指令環境, 175, 176, 179, 199, 211
 figure* 指令環境, 176
 \figurename, 177, 211, 252
 \fill, 113
 \filright, 293
 Flipo, Daniel, 219
 float (浮動版面), 162, 175, 179
 \flushcolumns, 127
 flushleft 指令環境, 111, 120
 flushright 指令環境, 111
 \fnsymbol, 124
 font, 67
 Computer Modern, 67
 Metafont, 67
 outline font, 67
 TrueType, 68
 Type 1 (PostScript), 67, 268
 font family (字體族), 69
 Roman family, 69
 sans serif (無裝飾邊), 69
 typewriter family, 69
 font mapping file, 274, 278
 font metrics, 273
 font series (字體序列), 69
 bold series, 69
 medium series, 69

 font shape (字形), 69
 italic, 69
 slanted, 69
 upright, 69
 font size (字級), 67, 247
 \fontdimen2, 273
 \fontfamily, 280, 291
 fontinst 巨集套件, 275, 276, 279
 Fontographer, 341
 \fontseries, 280
 \fontshape, 280, 291
 \fontsize, 23, 39, 76, 78, 124, 280
 footer (頁足), 86, 92
 footnote 計數器, 124
 \footnote, 123, 192, 251
 \footnotemark, 125
 \footnoterule, 125
 \footnotesep, 125
 \footnotesize, 75
 \footnotetext, 125
 \footrulewidth, 303
 fourier 巨集套件, 269
 frame 指令環境, 229
 \frame, 229
 \framebox, 121
 \frametitle, 229
 Franz, Melchior, 319
 \frontmatter, 104, 314
 ftnright 巨集套件, 128

 gather 指令環境, 157
 geometry 巨集套件, 283, 285, 319
 Ghostscript, 197, 199, 325
 gpdf, 211
 Grant, Michael C., 202
 graphics 巨集套件, 213

graphicx 巨集套件, 183, 188,
 199, 211, 213, 216, 224
 GSview, 37, 42, 199, 325
 Gustafson, Grant, 239
 Gutenberg, Johannes, 267
 \H{o} (ó 重音符號), 56
 \hangindent, 246
 Harders, Harald, 221
 header (頁眉), 86, 92
 \headheight, 85
 \headpagename, 252
 \headrulewidth, 302, 303
 \headtoname, 252
 Heath, Mark, 337
 \height, 120
 helvet 巨集套件, 268
 \hfill, 112, 121
 Hirata, Shunsaku, 325
 \hline, 163, 164, 170
 \hoffset, 86, 285
 \href, 327
 \hrulefill, 113, 292
 \hspace, 83, 112
 \hspace*, 83, 112
 HTML (Hypertext Markup Language), 323, 334
 \htmladdimg, 332
 \htmladdnormallink, 230, 333
 \htmlref, 333
 \huge, 75
 \Huge, 75
 \hyperlink, 230, 327
 hyperref 巨集套件, 230, 325,
 326, 328
 \hypertarget, 327
 hypertext links (超連結), 35,
 229
 hyphen (-), 見「減號」, 64
 \ifodd, 105
 ifthen 巨集套件, 249
 \ifthenelse, 250
 \ignorespaces, 250
 in (長度單位), 82
 \include, 238–240, 249, 315
 \includegraphics, 198, 199,
 201, 211
 \includeonly, 238–240, 315
 \indent (行首內縮), 109
 indent (行首內縮), 60
 \index, 305, 308
 \indexentry, 308
 \indexname, 252, 312
 initexmf, 279
 \input, 243, 248, 249
 \intextsep, 221
 \isodd, 250
 \it, 69, 70
 \item, 115–117
 itemize 指令環境, 115, 117, 229
 \itemsep, 117
 \itshape, 69
 JabRef, 257, 265
 Jeffrey, Alan, 275
 \jobname, 57, 111
 justification (右沿對齊), 110
 \kern, 352
 Kern, Uwe, 218
 \kill, 194
 Knuth, Donald Ervin (高德納),
 17, 67, 131, 267
 Krause, Dirk, 208
 \l (l), 57
 \L (L), 57
 \label, 102, 158, 178, 179, 230,
 313, 333
 \labelitemi, 116
 \labelitemii, 116
 \labelitemiii, 116
 \labelnumi, 117
 labels 巨集套件, 239, 240
 landscape, 85
 Lang, Russell, 199, 324
 \language, 66
 \leref, 91, 92, 246
 \large, 75
 \Large, 75
 \LARGE, 75, 287
 L^AT_EX, 245
 L^AT_EX2_ε, 88
 L^AT_EX2.09, 88
 \LaTeX (L^AT_EX), 56
 L^AT_EX2HTML, 324, 328, 334
 latexsym 巨集套件, 141, 144,
 186
 Lavagnino, John, 125
 \left, 150
 \left[, 149
 \leftmargini, 115, 117, 229
 \leftmarginii, 115, 117
 \leftmarginiii, 115
 \leftmark, 304
 Leichter, Terry, 182
 Lemberg, Werner, 337
 letter 文稿類別, 234, 252
 lettrine 巨集套件, 219
 \limits, 138
 \line, 222, 223
 \linespread, 76, 124, 229
 \linewidth, 85
 list 指令環境, 115, 117
 \listfigurename, 107, 252

`\listoffigures`, 107
`\listoftables`, 107
`\listtablename`, 107, 252
`\llap`, 見 `\rlap`, 181, 291
longtable 指令環境, 188, 192
longtable 巨集套件, 162, 188, 192
lscope 巨集套件, 188, 192
macros (巨集指令), 81, 245, 252
`\mainmatter`, 104, 314
makebb, 209
`\makebox`, 121
makebst 巨集套件, 265
makeeps, 16, 205, 207, 236, 335, 336
makefont, 338, 339
makeidx 巨集套件, 308
makeindex, 309
`\makeindex`, 308
`\maketitle`, 23, 89, 94, 303
`\marginpar`, 125, 126
`\marginparpush`, 85, 126
`\marginparsep`, 85, 126
`\marginparwidth`, 85, 126
`\markboth`, 103, 300
`\markright`, 103
math 指令環境, 133
math mode (數學模式), 132
`\mathbf`, 136
`\mathcal`, 136
Mathematica, 207
`\mathindent`, 87, 160
`\mathit`, 136
mathematical environment (數式環境), 132
mathpazo 巨集套件, 235, 268, 269, 291
mathptmx 巨集套件, 268
`\mathrm`, 136
`\mathsf`, 136
`\mathtt`, 136
Mattes, Eberhard, 277
`\mbox`, 105, 121, 151, 293
McDonnell, Rowland, 275
`\mdseries`, 69
`\medskip`, 29, 112, 245
memoir 巨集套件, 313
Metafont, 67, 267, 270, 272
`\midrule`, 166
MiKTeX, 27, 37, 205, 210, 336, 338
minipage 指令環境, 117, 120–122, 204
Mittelbach, Frank, 68, 126, 152
multicol 巨集套件, 126, 128
multicols 指令環境, 126, 128
`\multicolumn`, 126, 167, 170, 192
`\multiinclude`, 231
`\multiput`, 223
multirow 巨集套件, 182
`\multirow`, 182
multiline 指令環境, 156
`\mutlicolsep`, 127
mymacro 巨集套件, 77
natbib 巨集套件, 262, 264
`\newcolumnntype`, 180
`\newcommand`, 245
`\newenvironment`, 250
`\newline`, 見「`\` (換行)」, 110, 155
`\newpage`, 91, 104, 111, 314
`\newsavebox`, 123
`\newtheorem`, 152
`\newtheorem*`, 154
`\newtheoremstyle`, 155
NFSS (new font selection scheme), 68, 69, 76
中文字體指令, 71
Niepraschk, Rolf, 210
Nikon D2X 相機, 313
`\nocite`, 263
`\nocite*`, 263
`\noindent`, 60, 109
`\nolimits`, 138
`\normalfont`, 292
`\not`, 140
`\notag`, 156
`\o` (ø), 57
`\o`(Ø), 57
`\oddsidemargin`, 85
`\oe` (œ), 57
`\OE`(Œ), 57
`\onecolumn`, 317
Oostrum, Piet van, 182
`\opening`, 233
`\oval`, 222, 223
`\P` (¶), 57
package (巨集套件), 81, 87
page 計數器, 104
page style (頁面格式), 86, 298
`\pagebreak`, 127, 190
`\pagecolor`, 218
`\pagenumbering`, 104
`\pageref`, 102, 178, 179, 250
`\pagestyle`, 102, 103, 298
empty 選項, 102
headings 選項, 102
myheadings 選項, 102
plain 選項, 102
`\paperheight`, 85, 239
`\paperwidth`, 85
`\par`, 59, 109, 111, 119

`\paragraph`, 96
`\parbox`, 117, 120, 121, 204, 211
`\parindent`, 109, 119, 154, 246
`\parskip`, 109
`\part`, 96
`\partname`, 252
Patashnik, Oren, 253
`\pause`, 35, 36, 229–232
pc (長度單位), 82
PDF, 35, 42, 198, 227, 228, 232, 323, 334
 bookmarks (書籤), 230, 327
 bounding box, 210
 epstopdf, 210
 以 Distiller 輸出, 325
 以 dvipdfmx 輸出, 325
 以 pdf \LaTeX 輸出, 35, 325
 以 ps2pdf 輸出, 35, 325
 超連結, 326
 搜尋中文, 42, 326
 與 HTML 比較, 323
 與 PostScript 之關係, 40
pdf \TeX , 227, 232, 325
perl, 205, 328, 330, 335
picture 指令環境, 221, 224
Piff, Mike, 236, 241
`pltotf`, 276, 277
`\pmod`, 145
portrait, 85
PostScript, 197–199
 EPS, 199
 PostScript (Type 1) 字體, 267, 270, 281
 PostScript 印表機, 198
 PostScript 印表機驅動程式, 204
`\pounds` (£), 55, 57
Power Point, 227
 preamble (全文設定區), 73, 76, 84, 86, 87, 96, 98, 100–103, 106, 108, 124, 138, 140, 156, 217, 224, 245, 248, 250, 268, 269, 279, 284, 299–301, 304, 336, 348
`\prefacename`, 252
`\printindex`, 308
printing points (點), 82
`\proofname`, 154
`\ps`, 234
ps2pdf, 35, 42, 211, 325
ps2up, 42
ps4pdf 巨集套件, 210
psbook, 322
`\PSforPDF`, 211
psfrag 巨集套件, 202–204
`\psfrag`, 202–204, 210, 211
PSNFSS 巨集套件, 268, 270
psnup, 320, 321
pst-tree 巨集套件, 224
pstcol 巨集套件, 224
PSTricks 巨集套件, 224, 225
psutils, 320, 322
pt (長度單位), 82
`\put`, 222, 223
pxfonts 巨集套件, 269
`\qbezier`, 223
`\qqquad`, 112
`\quad`, 100, 112, 158
quotation 指令環境, 114
quote 指令環境, 73, 114
`\raggedcolumns`, 127
`\raggedleft`, 110
`\raggedright`, 110, 293
Rahtz, Sebastian, 188, 239, 268, 326
`\raisebox`, 64, 122, 183, 336
Rancilio Audrey 咖啡機, 18
`\rangle`, 183
Reckdahl, Keith, 198
`\ref`, 102, 158, 178, 179, 230, 313, 333
`\reflectbox`, 215, 216
`\refname`, 252, 254
`\renewcommand`, 76, 96, 245, 246
`\renewenvironment`, 250
`\resizebox`, 215, 216
`\resizebox*`, 216
RGB, 218
`\right`, 150
`\right]`, 149
`\rightmark`, 304
`\rightskip`, 126
`\rlap`, 見 `\llap`, 181
`\rm`, 69, 70
`\rmfamily`, 69
Rokicki, Tom, 320
`\roman`, 99, 125
`\Roman`, 99, 125, 292
`\rotatebox`, 188, 192, 214
rotating 巨集套件, 192
`\rowcolor`, 185, 186
`\rule`, 83, 122, 125
running-head, 97, 292
Ryu, Young, 269
`\S` (§), 57, 100
`\samepage`, 105
sans serif (無裝飾邊字形), 見 serif, 78
`\savebox`, 123
`\sc`, 69, 70
`\scalebox`, 215, 216

Schöpf, Rainer, 68, 152

`\scriptscriptstyle`, 160

`\scriptsize`, 75

`\scriptstyle`, 139, 148, 160

`\scshape`, 69

`\section`, 72, 89, 96, 228, 286

`\section*`, 97

`\sectionmark`, 300, 301

`\seename`, 252

`\selectfont`, 280

serif (裝飾邊字形), 見 sans serif, 78

`\setcounter`, 103, 124, 158, 251

`setcwtex`, 54

`\setlength`, 76, 82

`setwed54`, 51

`\sf`, 69, 70, 287

`\sffamily`, 69, 292

`\shortstack`, 222, 223

`\sidewaysfigure`, 188

`\sidewaystable`, 188

`\signature`, 234

`\sl`, 69, 70

`\slshape`, 69

`\small`, 75, 94

`\smallskip`, 29, 112

`\smallskipamount`, 112

Sommerfeldt, Axel, 177

`\special`, 197

split 指令環境, 156

`\sqrt`, 135

`\ss` (β), 57

`\stackrel`, 147

`\subparagraph`, 96

`\subsection`, 89, 96, 228

`\subsection*`, 97

`\subsectionmark`, 300, 301

`\subsubsection`, 96

`\swapnumber`, 154

`\t{oo}` ($\text{\textcircled{oo}}$ 重音符號), 56

tabbing 指令環境, 162, 193, 195

`\tabbingsep`, 194

`\tabcolsep`, 170

table 指令環境, 175, 179, 211

table* 指令環境, 175

`\tablename`, 106, 177, 252

tablenotes 指令環境, 169

`\tableofcontents`, 91, 106, 294, 296, 315

tabular 指令環境, 162, 164, 173

`\tabularcolumn`, 174

tabularx 巨集套件, 162, 173, 174

`\tag`, 156

`\tag*`, 156

Tantau, Till, 227

`\tb` (排版迴歸式), 138, 247

testcwi, 49

TeX, 18, 245

TeX capacity exceeded (錯誤訊息), 343, 348

tex2xtc, 107, 309

`\textbf`, 69

`\textcolor`, 187, 217, 221, 229, 289

`\textheight`, 85, 86, 283, 314

`\textit`, 69

`\textmd`, 69

textmerg 巨集套件, 236, 239

`\textrm`, 69

`\textsc`, 69

`\textsf`, 69

`\textsl`, 69

`\textstyle`, 160

`\texttt`, 69

`\textup`, 69

`\textwidth`, 82, 85, 86, 283

Thành, Hàn Thê, 325

`\thanks`, 94

`\thechapter`, 292

`\thecontentslabel`, 297

`\thefootnote`, 124

theorem 巨集套件, 152, 155

`\theoremstyle`, 154

`\thesection`, 98, 292, 301

`\thesubsection`, 98

`\thetitle`, 100

`\thicklines`, 223

`\thinlines`, 223

`\thispagestyle`, 91, 102, 103, 105, 303, 304

threeparttable 指令環境, 169

threeparttable 巨集套件, 169

times 巨集套件, 268

`\tiny`, 75

`\title`, 23, 73, 89, 92, 94, 229

`\titlecontents`, 296, 298

`\titlecontents*`, 298

`\titleformat`, 286, 295

`\titlelabel`, 100

`\titleline`, 287

titlepage 指令環境, 95

`\titlerule`, 287

titlesec 巨集套件, 89, 94, 98–100, 283, 286, 295

`\titlespacing`, 286, 295

titletoc 巨集套件, 295, 298

`\tnote`, 169

tocdepth 計數器, 106

`\today`, 56, 111, 234

`\topcaption`, 177

`\topmargin`, 85

`\toprule`, 166

Total commander, 52

`\totalheight`, 120
 TrueType 字體, 270
`\tt`, 69, 70
`tft2pfb`, 281, 337
`tft2pt1`, 337
`\ttfamily`, 69
`twocolumn` 選項, 87
`\twocolumn`, 317
`\twoside`, 85
`txfonts` 巨集套件, 269
`type1cm` 巨集套件, 78

`\u{o}` (ö 重音符號), 56
 Umeki, Hideo, 283
`\unboldmath`, 137
`\underline`, 71, 146
`unicode`, 337
`\unitlength`, 222
`\upshape`, 69
`url` 巨集套件, 58, 261
`\url`, 58
`\usebox`, 123
`\usepackage`, 88

`\v{o}` (ö 重音符號), 56
 van Oostrum, Piet, 298
`\vector`, 222, 223
`\verb`, 128, 129
`verbatim` 指令環境, 128, 129
`Verbatim` 指令環境, 128, 130
`verse` 指令環境, 114
`\vfill`, 111, 112
 Visio, 204, 206
`\vline`, 170, 172
`\voffset`, 86, 285
`vptovf`, 276, 277
`\vrule`, 172
`\vspace`, 83, 112
`\vspace*`, 83, 112

 Weeks, Andrew, 337
`\width`, 120
 Wilson, Peter, 286
 WinEdt, 37
 功能鍵, 41
 功能鍵設定, 61
 輸入數學符號, 145
`wrapfig` 巨集套件, 220
`wrapfigure` 指令環境, 220

`xcolor` 巨集套件, 218, 230, 290
`xmpmulti` 巨集套件, 229

 YAP, 37, 40, 42

`\Z` (中文與數字間距), 179
 Zandt, Timothy van, 128, 224
 Zapf, Hermann, 269

 3 劃
 大宗信函, 236
 工作檔案夾, 38, 44, 52, 107, 255

 4 劃
 中文字體指令, 71, 352
 `\ctxfdef`, 25, 73, 74, 89, 91,
 101, 114, 129, 178, 287,
 301
 字距, 352
 字體變形, 73, 272, 275
 新式指令, 72
 舊式指令, 72, 350
 分式, 134
 引述 (cross-reference), 101, 102,
 158, 178, 179, 313, 326
 `\label`, 101, 158, 178, 179
 `\pageref`, 101
 `\ref`, 101, 158, 178, 179

 圖表, 178
 數學式, 158
 文字方塊 (box)
 LR 方塊, 121
 段落方塊, 121
 線條方塊, 121
 文字編輯軟體
 Crimson editor, 42
 WinEdt, 37, 40
 功能鍵設定, 44
 輸入中文標點符號, 45
 文稿結構 (document structure),
 81, 86, 89
 文稿類別 (document class), 86,
 89
 amsart, 86
 amsbook, 86
 article, 39, 86, 88, 89
 beamer, 35, 227
 book, 86, 92, 96
 cwlletter, 236
 letter, 86, 232, 234
 report, 86
 slides, 86
 文獻資料格式, 254
 方塊 (box), 121

 5 劃
 巨集指令 (macros), 67, 74, 77,
 123, 138, 245, 248, 252,
 313, 315, 317, 318
 中文, 248, 315
 巨集套件, 81, 87, 88
 amsmath, 137, 152, 155, 158
 amssymb, 131, 140, 141
 amsthm, 152, 155
 answers, 241, 243
 appendix, 108

array, 87, 162, 163, 169, 172
 avant, 268
 babel, 56, 65, 66
 beamer, 35, 227, 232
 bibentry, 264
 bm, 137
 bmeps, 208
 bookman, 268
 booktabs, 162, 163, 166, 167
 caption, 177, 190
 ccfonts, 269
 chancery, 268
 charter, 268
 chngpage, 286, 317
 color, 184, 187, 216, 218, 224, 289
 colortbl, 185, 187
 comment, 128
 contour, 221
 courier, 268
 crop, 319
 cwtex, 330
 dcolumn, 162, 165, 179, 182
 endnotes, 125
 fancyhdr, 87, 93, 102, 106, 283, 298, 304
 fancyvrb, 128, 130
 fontinst, 275, 276, 279
 fourier, 269
 ftnright, 128
 geometry, 283, 285, 319
 graphics, 213
 graphicx, 183, 188, 199, 211, 213, 216, 224
 helvet, 268
 hyperref, 230, 325, 326, 328
 ifthen, 249
 labels, 239, 240
 latexsym, 141, 144, 186
 lettrine, 219
 longtable, 162, 188, 192
 lscape, 188, 192
 makebst, 265
 makeidx, 308
 mathpazo, 235, 268, 269, 291
 mathptmx, 268
 memoir, 313
 multicol, 126, 128
 multirow, 182
 mymacro, 77
 natbib, 262, 264
 ps4pdf, 210
 psfrag, 202–204
 PSNFSS, 268, 270
 pst-tree, 224
 pstcol, 224
 PSTricks, 224, 225
 pxfonts, 269
 rotating, 192
 tabularx, 162, 173, 174
 textmerg, 236, 239
 theorem, 152, 155
 threeparttable, 169
 times, 268
 titlesec, 89, 94, 98–100, 283, 286, 295
 titletoc, 295, 298
 txfonts, 269
 type1cm, 78
 url, 58, 261
 wrapfig, 220
 xcolor, 218, 230, 290
 xmpmulti, 229
 正文方塊, 85
 目錄, 106, 295
 立體陰影 (dropping shadows), 221
 6劃
 全文設定區 (preamble), 73, 76, 84, 86, 87, 96, 98, 100–103, 106, 108, 124, 138, 140, 156, 217, 224, 245, 248, 250, 268, 269, 279, 284, 299–301, 304, 336, 348
 多欄位版面, 87, 126, 128
 印表機
 PostScript 印表機驅動程式, 204
 字型
 加裝中文字型, 54
 字型度量 (font metrics), 273
 字型度量檔, 275, 341
 字型對應檔 (font mapping file), 274, 278
 字級
 相對大小指令, 75
 \footnotesize, 75
 \huge, 75
 \Huge, 75
 \large, 75
 \Large, 75
 \LARGE, 75
 \normalsize, 75
 \scriptsize, 75
 \small, 75
 \tiny, 75
 字級 (font size), 67, 74, 247
 相對大小指令之字級指令, 75
 字體, 67
 avant 巨集套件, 268

- bookman 巨集套件, 268
- ccfonts 巨集套件, 269
- chancery 巨集套件, 268
- Charter, 268
- charter 巨集套件, 268
- Computer Modern, 270
 - PostScript 格式, 270, 272
- courier 巨集套件, 268
- fourier 巨集套件, 269
- Garamond, 270
- helvet 巨集套件, 268
- Helvetica, 268, 278
- mathpazo 巨集套件, 268
- mathptmx 巨集套件, 268
- Mathtime, 270
- Metafont, 267
- outline font, 67
- Palatino, 235, 269, 275
- PostScript, 267, 281
- pxfonts 巨集套件, 269
- small capital, 280
- Times, 269
- TrueType, 68, 272, 281, 335
- txfonts 巨集套件, 269
- Type 1 (PostScript), 67, 68
- unicode, 337
- Utopia, 268
- 中文字排序, 342
- 中文字體, 272, 338
- 中文字體名稱, 338
- 字形 (font shape), 69
- 字級, 74
- 字體序列 (font series), 69
- 字體族 (font family), 69
- 設計尺寸 (design size), 74, 82
- 造字, 341
- 描點字型, 270
- 描邊字型, 270
- 數學字體, 136, 268, 270, 272
- 數學粗體, 137
- 字體大小, 見「字級」, 23
- 字體指令, 279, 280
 - \ctxfoff, 333
 - \ctxfon, 333
 - 中文字體指令, 71, 74
 - 宣告字體指令, 69
 - 英文字體指令, 68, 71
 - 設定全文之中文字體, 73, 74
 - 數學字體指令, 136
 - 標準字體指令, 69
 - 調整中文字高低位置, 352
- 灰階 (grayscale), 216
- 考題與解答, 241, 243
- 行列式, 148, 151
- 行長, 79, 85
- 行首內縮 (indent), 60, 109
- 行距, 23, 79
 - \baselineskip, 76
 - \linespread, 229
 - 太小, 79
 - 變更, 76, 78
- 7 劃
- 希臘字母符號, 135
- 批次檔, 38, 49, 205, 211, 275, 335, 338
 - autoexec.bat, 50
- 投影片, 86, 227, 232
 - PDF, 227
- 李君宇, 337
- 李果正, 337
- 8 劃
- 兩欄式版面, 286
- 函數符號, 137
- 固定長度 (fixed length), 見「彈性長度」, 82
- 固定格式標籤, 239, 240
- 居中編排, 110
- 版面
 - 單雙頁版面位置差異, 86, 87, 285, 299
- 版面方向
 - 風景模式 (landscape), 85
 - 畫像模式 (portrait), 85
- 表格
 - array 巨集套件, 162, 163, 169, 172
 - array 指令環境, 162
 - booktabs 巨集套件, 162, 163, 166
 - dcolumn 巨集套件, 162
 - tabbing 指令環境, 162, 193, 195
 - tabular 指令環境, 162
 - tabularx 巨集套件, 162
 - tabularx 指令環境, 173, 174
 - 小數點對齊, 179
 - 加大行距, 164
 - 加入斜線, 183, 184
 - 加入短直線, 172
 - 居中, 164
 - 旋轉, 187, 188
 - 註解, 168, 169
 - 超大型, 187, 192
 - 超寬表格, 188
 - 數字上下對齊, 165
 - 標題, 175, 178
 - 欄位寬度, 173
- 長度單位, 82
 - cm, 82
 - em, 82
 - ex, 82

- in, 82
- pc (pica), 82
- pt (point), 82
- 附錄, 107
- 9劃
- 信函, 86, 232, 233, 240
 - 大宗信函, 236, 239
 - 信頭標識, 234, 239
- 指令 (command)
 - 強制變數 (mandatory argument), 83
 - 選項變數 (optional argument), 83
- 指令環境 (environment), 73, 84
- 段落間距 (\parskip), 109
- 界限符號 (delimiter), 149, 183
- 美國數學學會, 131, 140
- 計數器 (counter), 251
 - addtocounter, 251
 - chapter, 251, 292
 - enumi, 251
 - enumii, 251
 - enumiii, 251
 - enumvi, 251
 - equation, 158, 251
 - figure, 251
 - footnote, 124, 251
 - mpfootnote, 251
 - page, 103, 251
 - paragraph, 251
 - part, 251
 - section, 98, 251
 - subparagraph, 251
 - subsection, 98, 251
 - subsubsection, 251
 - table, 251
- 重音符號 (accents), 56, 195
 - 數學, 145
- 頁足 (footer), 86, 92, 105, 302, 304
- 頁眉 (header), 85, 86, 92, 105, 302, 304
- 頁眉之章節標題
 - 中文字亂碼問題, 102, 300
- 頁面格式 (page style), 86, 102, 298, 304
- 頁碼, 87, 102, 104, 298, 304, 314
- 頁碼字體, 104
- 頁碼計數器, 104
- 10劃
- 展示數式 (display formula), 132
- 浮動版面 (float), 162, 175, 211
 - figure 指令環境, 162, 175
 - table 指令環境, 162, 175
- 矩陣, 148, 151
- 索引, 252, 305–312, 317
 - 特殊符號, 306, 308
 - 排序, 312
 - 標識索引名詞, 305
- 紙張
 - a4paper, 23, 39, 84
 - a5paper, 84
 - b5paper, 84
 - executivepaper, 84
 - legalpaper, 84
 - letterpaper, 84, 239, 240
- 紙張尺寸, 84
- 翁鴻翎, 17, 335, 337
- 迷你版面, 117, 123
- 迴歸式, 138, 247
- 11劃
- 參考文獻, 20, 92, 253, 265
 - 中文排序, 255
 - 文獻格式, 265
 - 文獻資料檔, 253, 255–257, 261
- 基線 (baseline), 118, 149
- 常見問題
 - PDF 之超連結, 228, 325
 - PDF 搜尋中文, 54, 326
 - pdflatex 不接受 \psfrag 指令, 204
 - .sty not found, 52
 - TeX FAQ, 55
 - YAP 中文顯示, 54
 - YAP 無法顯示 \psfrag 之替代字串, 204
 - 中文字大小不一, 351
 - 引用 EPS 圖形, 但圖形無法顯示, 204
 - 如何加裝字型, 54
 - 如何取消 PDF bookmarks (書籤), 327
 - 如何設定 WinEdt 功能鍵, 51
 - 如何造字, 335
 - 安裝, 49
 - 使用 Windows 字型, 337
 - 使用舊版字型, 54
 - 參考文獻大寫字母變成小寫, 261
 - 參考文獻加入目錄, 91
 - 排版研究生論文, 89
 - 排版超長表格, 188
 - 排版超寬表格, 249
 - 救回原稿, 349
 - 設定 path 環境變數, 50
 - 設定章節之中文字體, 100
 - 章節標題加入目錄, 91, 106
 - 章節編號以國字排版, 289
 - 解讀排版訊息, 26, 27, 343
 - 標點符號如何居中排版, 19

- 調整列印版面, 86
- 輸入標點符號, 64
- 輸出 EPS 圖形, 204
- 變更頁碼, 102
- 常見排版錯誤
 - 中文稿輸入, 60
 - 中英文字體搭配, 98
 - 行距太小, 23, 79
 - 表格加入垂直線, 161
 - 展示數式未加入標點符號, 133
 - 排版符號, 55
 - 換行與空格, 59
 - 減號與破折號, 64
 - 圖表位置, 31, 175
 - 數式內排版普通文字, 150
 - 數學式與上下文間距, 133
 - 選用不當字體, 80
- 彩色圖文, 216, 218
- 排版, 56
 - 多欄, 126, 128
 - 兩欄, 109, 126, 128, 317
 - 直排, 73
 - 研究生論文, 89, 92
 - 書籍, 314, 319
 - 單欄, 317
 - 幕前排版, 37
 - 幕後排版, 37
 - 德文, 56
 - 歐洲國家文字, 65
 - 橫排, 73
- 排版步驟, 39, 40
- 排版訊息, 26, 27, 343
- 旋轉文字或圖表, 214
- 條列指令環境, 115, 117
 - description, 115
 - enumerate, 115
 - itemize, 115
 - 內縮距離, 117
- 軟體安裝
 - cwTeX, 47
 - TeX2HTML, 328
 - Linux 版本, 54
 - perl, 328
- 軟體更新
 - MiKTeX, 53
- 軟體設定
 - MiKTeX, 52
 - WinEdt, 51
 - Total commander, 52
- 造字, 335, 337
 - 由 TrueType 字型輸出 Type 1 字型, 337
 - 圖形字, 335
- 章節標題, 96, 99, 101, 283, 286
 - 彩色, 289
- 12 劃
 - 換行指令, 29, 61, 62, 95, 110, 119, 148, 168, 190, 214, 292
 - \\, 110
 - \newline, 110
 - 換頁, 104, 106, 306
 - 期望值, 138
 - 減號 (—), 見 hyphen, 64
 - 裁切記號 (cropmark), 319
 - 註解, 123, 126
 - \fnsymbol, 124
 - 分隔線 (\footnoterule), 125
 - 行距調整, 124
 - 表格, 169
 - 間距 (\footnotesep), 125
 - 註銷指令 (%), 57
 - 超連結 (hypertext links), 35, 229
- 13 劃
 - 填入直線 (\hrulefill), 113
 - 填入細點 (\dotfill), 113
 - 微分符號, 134
 - 照列原文 (verbatim), 128, 130
- 14 劃
- 圖形
 - bounding box, 199, 320
 - EPS, 199, 204, 211, 320
 - Excel, 207
 - 引用 JPEG 圖形, 208
 - 引用 PDF 圖形, 198, 199, 209
 - 引用 PNG 圖形, 208
 - 引用 PostScript 圖形, 198, 207
 - 引用描點圖形, 208
 - 加入中文或數式, 201, 204
 - 常見問題, 204
 - 彩色, 216, 218
 - 旋轉, 214
 - 描點圖形, 197
 - 描邊圖形, 197
 - 圖檔規格, 197, 199
 - 標題, 199, 211
 - 標識數學式, 201, 204
 - 樹狀圖, 223
 - 縮放, 200, 215
 - 轉換格式, 208
 - 繪圖軟體, 206, 207
 - 顯示問題, 204, 207, 213, 216
- 圖表標題, 176, 252
- 圖檔規格
 - EPS, 198, 211
 - JPEG, 197, 198, 230
 - PDF, 197, 198
 - PNG, 197, 230
- 幕前排版, 17

- 幕後排版, 17
- 摘要, 96
- 網路出版, 323, 334
- 15 劃
- 彈性長度 (rubber length), 見
「固定長度」, 82
- 數字格式
- alph, 104
 - \alph, 99
 - Alph, 104
 - \Alph, 99
 - arabic, 104
 - \arabic, 99
 - roman, 104
 - \roman, 99
 - Roman, 104
 - \Roman, 99
- 數式環境, 132
- 數學式
- 下標, 134
 - 上標, 134
 - 引述數式, 158
 - 左下標, 151
 - 多行數學式, 155, 158
 - 定義與定理, 152, 155
 - 居中或靠左, 160
 - 界限符號, 149
 - 矩陣與行列式, 148, 151
 - 迴歸式, 138
 - 粗體符號, 137
 - 連分式, 159
 - 畫底線, 146
 - 証明, 154
 - 間距調整, 158, 159
 - 與上下文間距, 133
 - 數學字體指令, 136
- 數學花體字 (script letters), 136
- 編號, 158, 159
- 調整符號大小
- \displaystyle, 160
 - \scriptscriptstyle, 160
 - \scriptstyle, 160
 - \textstyle, 160
- 調整間距
- !\, 159
 - \,, 159
 - \:, 159
 - \;, 159
- 雙元運算符號, 143
- 數式內排版中文, 150
- 數式內排版英文, 150
- 數學重音符號, 145
- 數學符號
- AMS 其他符號, 145
 - AMS 負雙元關係, 143
 - AMS 箭頭符號, 141
 - AMS 雙元運算, 144
 - AMS 雙元關係, 142
 - \because (∴), 140
 - \therefore (∴), 140
 - 加總函數, 138
 - 希臘字母符號, 135
 - 其他, 145
 - 函數, 137
 - 相對關係, 139, 141
 - 符號上下推疊, 147
 - 粗體數學符號, 136
 - 連續點, 148
 - 期望值, 138
 - 開根號, 135
 - 數學重音, 145
 - 箭號, 139
- 箭頭符號, 140
- 積分符號, 138
- 雙元運算, 144
- 變異數, 138
- 數學符號間距調整, 159
- 數學模式, 132
- 標點符號, 19, 59, 61, 62, 133, 273
- 中文, 45, 258, 261
 - 全型輸入, 62
 - 居中排版, 19
 - 英文右單引號 ('), 63
 - 英文右雙引號 ("), 63
 - 英文左單引號 (‘), 63
 - 英文左雙引號 (‘), 63
 - 英文標點符號, 62
 - 破折號, 63
 - 換行, 29
 - 與註解指令之相對位置, 123
 - 數學式, 133
 - 輸入, 45, 62
 - 避頭點, 65
- 標題
- 序文與索引, 294
 - 章節, 283, 286, 315
 - 圖表, 199
- 標題格式 (\titleformat)
- block, 287
 - display, 287, 290
 - drop, 287
 - frame, 287
 - hang, 287, 290
 - leftmargin, 287
 - rightmargin, 287
 - runin, 287
 - wrap, 287
- 線條方塊 (\rule), 122
- 調整行文間距, 111

靠右編排 (flushright), 110

靠左編排 (flushleft), 110

16 劃

樹狀圖 (trees), 225

輸入文稿

中文稿, 60, 65

英文稿, 59, 60

數學文稿, 133

標點符號, 62

隨文數式 (in-text formula), 132,

133

17 劃

儲存方塊 (\sbox), 123

檔案搜尋, 52, 53

LaTeX2HTML, 331

MiKTeX, 52, 53, 331, 336

文獻資料檔, 256

圖形檔, 236

縮放文字圖表, 215

避頭點, 65

點 (printing point), 82

19 劃

邊註 (marginal notes), 85, 123,

125, 126

23 劃

變異數, 138