

ChiTeX (中文 TeX/LaTeX) 簡介*

陳弘毅[†]

摘要

TeX/LaTeX 為數理文書排版的強大功能軟體，而 chiTeX/chiLaTeX 則為架構在 TeX/LaTeX 上之 '中文 TeX/LaTeX'，它可輸入、編譯含或不含中文之文件，係由本人所發展，有 unix、Windows 等版本。

目錄

1 如何安裝 chitex	3
2 TeX/LaTeX 簡介	4
3 chitex 簡介	5
4 利用 chitex 編寫數理文件	7
5 LaTeX 文件的一般架構	9
5.1 基本的架構	9
5.2 標題頁(Title Page)	13
6 LaTeX 文件的語法(二)	15
7 如何打數學式子	17
8 章節指令	18
9 字形指令	20

*本文件利用 χ TeX 編寫

[†]國立中央大學數學系 (yih@math.ncu.edu.tw)

10 再談數學式子(一)	23
11 再談數學式子(二)—陣列、矩陣等	30
12 再談數學式子(三)—條件式、多行數學式子等	34
13 引述、條列、詩文環境指令	36
14 行距指令	41

1 如何安裝 chitex

如何安裝 chitex:

(A) Windows 版: 到

`ftp://ftp.math.ncu.edu.tw/chitex/Win2000-XP-Vista`

抓取其中整個子目錄: `ps635Mk27b41`

然後在目錄 `ps635Mk27b41` 中, click `setup(.exe)`”

(B) Unix 版: 到

`ftp://ftp.math.ncu.edu.tw/chitex/unix`

抓取其中整個子目錄: `6.1.2p15a38`

然後在目錄 中, run: `./setup`

(Before run 'setup', make 'setup' executable)

2 TeX/LaTeX 簡介

- (1) TeX 為美國 Stanford 大學教授 Knuth 在 1978 年所創的一套專門為處理 '數理方程式' 的排版系統。一直到 1985 年始移植到個人電腦上。
- (2) TeX 是一套容易使用的幕後排版系統, 對於需處理數理方程、科學符號的文章與科學論文, 它提供了精確且精美的品質。只要下幾個簡單指令, 就可以得到一篇含有複雜數學式子的文件, 而且自動幫你 '左對齊'、'右對齊', 自動分頁, 加 '註解', ...
- (3) LaTeX 為美國 Laslie Lamport 在 1985 年所發展, 它是架構在 TeX 上的一套可提供各種 '文件格式' 的排版軟體。其 '文件格式' 含有:
article.cls (普通文件), report.cls(報告), book.cls(書),
letter.cls(信件), slides.cls(投影片)等。
- (4) 今天, 使用 TeX/LaTeX 已成為全世界大部份數理論文的標準。許多論文期刊都要求作者必須使用 TeX/LaTeX 編寫論文。

3 chitex 簡介

簡單地說, chitex 與 chilatex 分別是中文 (Big5碼繁體字及 GB碼簡體字) 的 plain TeX 與 LaTeX。可用於 unix 系統, 與 Windows 系統。

一般 chitex 指的就是包含 chitex 及 chilatex 的泛稱。在 unix 與 Windows 上的命令或視窗功能, chitex 都能自動編譯 plain TeX 與 LaTeX 文件。

使用 chitex 很容易, 會用英文 TeX/LaTeX 就可立即使用 chitex。最簡單的使用 chitex, 只要在文件中, 輸入中文就可。例如, 下面是一 plain TeX 文件:

```
%-----  
    這是一個 plain TeX 文件  
  
    數學式子:  $x^2+y^2=1$  為一圓。  
  
    \bye  
%-----
```

存檔編譯(見後面說明)後就可得到:

```
    這是一個 plain TeX 文件  
    數學式子:  $x^2 + y^2 = 1$  為一圓。
```

若要多一些對中文的控制, 也只要知道幾個與中文字型有關的指令就可以。

下面列出幾個使用中文的注意事項與中文字型有關的指令：

- (1) 文件中的中文與英文之間，最好有一空白，才可使中英文夾雜時較勻稱。
- (2) 預設的中文字形為：
新細明體
(即不下中文字型指令時，等於已下字形指令 `\mingu`)
(TeX/LaTeX 的指令都以 `\` 開頭)
- (3) 立即可使用的 `chitex` 的中文字型指令：
有多種字形可立即可使用，例如：
`\kaiu`, `\mingu`, `\KAI`, `\LI`, ...
詳見：中文字形指令
- (4) 字型放大指令：
有方便的有段與無段之字型放大指令，見 中文字形指令
- (5) 仍有其他方便的指令，將在後面說明。

4 利用 chitex 編寫數理文件

下面介紹 chitex 的使用 (chitex 包含 chilatex)。

從最簡單的開始，如果要得到一篇如下的測試文章：

1. 設 $F(x) = \int_0^x f(t) dt$ 求: $F'(x)$
2. 試求積分 $\int \frac{x^2+x+1}{x^2-1} dx$

可經下列步驟得到：

1. 首先用文書編輯，編寫一檔案內容如下：

```
%-----  
1. 設  $F(x)=\int_0^x f(t)\,dt$  求:  $F'(x)$   
  
2. 試求積分  $\int \frac{x^2+x+1}{x^2-1}\,dx$   
\bye  
%-----
```

以上是用 plain TeX 文件編寫，下面則是用 LaTeX 文件編寫，也可有同樣結果。因 LaTeX 功能豐富且較方便，**以後都將全部介紹 LaTeX 文件的編寫語法。**

```
%-----  
\documentclass[12pt]{article} %每一 LaTeX 文件都要  
%以指令 \documentclass  
%開頭  
  
\begin{document}
```

1. 設 $F(x) = \int_0^x f(t) dt$ 求: $F'(x)$

2. 試求積分 $\int \frac{x^2+x+1}{x^2-1} dx$

`\end{document}`

%-----

(TeX/LaTeX 語法說明 :

- 在 $\$$ 與 $\$$ 之間的, 是數學式子的內容。凡是數學式子都要以一對 $\$$ 圍起來。

- `\int` 編譯後會得到 \int , 以 `\` 為開頭者為 TeX/LaTeX 的指令

- `_ ^` 的作用是分別得到 "上標及下標", 例如:

`x^2, x^{10}, a_1, a_{10} \int_0^x`

可分別得到:

$$a^2, x^{10}, a_1, a_{10}, \int_0^x$$

)

2. 其次將上述檔案存檔

例如存成檔案 `testp1.tex`。TeX/LaTeX 文件檔的副檔名內定為 `tex`, 因此一般副檔名皆為 `tex`, 當然有時也可用其他的。(若用 `\include` 指令讀進的 LaTeX 文件檔副檔名則一定須為 `tex`)

3. 然後利用 `chitex` 的編譯預視所存的上述文件檔

- 若用 Windows 版的 `chitex` 只要 ChiTeX 視窗中之功能表, 點取選項 (或 click 工具列上的圖示) 就可, 例如: 可在;



中可 click: 標有:

TeX

psv

的圖示，就可編譯預視。

(實際上 chitex 在背後自動進行:

編譯產生 testp1.dvi，再用 dvips 產生 testp1.ps，

再用 gscview 預視 testp1.ps

)

- 若用 Windows 版的 chitex 本項目可不看。

若用 unix 版的 chitex 就要利用 chitex 的編譯，就可產生 'dvi' 檔。再在 unix 中可用 xdvi 或 ghostview 預視，最後用 dvips 產生 'ps' 檔，且利用 lpr 列印。如下:

- (1) 先用 chitex 編譯如下:

chitex testp1 (或 chitex testp1.tex)

就會產生 'dvi' 檔: testp1.dvi

- (2) 預視有兩種方法，利用 xdvi 或 ghostview:

- (a) 利用 xdvi 可下命令如:

xdvi testp1 (或 xdvi testp1.dvi)

- (b) 利用 ghostview, 須先經 dvips 產生 'ps' 檔, 命令如下:

- (i) 先由:

dvips testp1 (或 dvips testp1.dvi)

產生 'ps' 檔: testp1.ps

- (ii) 再用 ghostview 預視如下:

gv testp1.ps

5 LaTeX 文件的一般架構

5.1 基本的架構

LaTeX 文件的內容必須有如下的基本的架構:

```
%-----  
\documentclass[12pt]{article} %每一 LaTeX 文件都要以指令  
% \documentclass 開頭
```

```

% article 為一文件格式，可換
% 為其他格式如: report, book,...等

\usepackage{...}
\usepackage{...}
...
\input{...}           % 可有可無
                      % (\documentclass 與 \begin{document}
                      % 之間是為 preamble
                      % (LaTeX 文件之 "前文"))

\begin{document}
...
...                   % \begin{document} 與 \end{document}
                      % 是為 LaTeX 文件之本體

\end{document}
%-----

```

今再詳細說明如下;

LaTeX 文件必須:

- (a) 以 `\` 為開頭者為 TeX/LaTeX 的指令, 例如: `\documentclass`, `\usepackage`, `\begin`, `\int` 等皆是。
- (b) `%` 用來當做 "註解、說明" 的開始, 一行中 `%` 及 `%` 之右邊的所有文字皆為 "註解" 編譯不會理會它, 預視與印出的文件內容不會包含這些。
- (c) 以 `\documentclass` 開始一篇文件, 用來選擇 '文件格式'. 例如, 上一節之 `test1.tex` 選擇 `article` 的格式(也就是選了 `article.cls` 這檔案所定的格式). 在該檔案中, 以

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

開始, 表示選擇 `article` 之文件格式且選擇選項 `12pt`。文件格式有多種, 常用的有:

`article`, `report`, `book`, `letter`, `slides`, `amsart`, `amsbook`

等, 而選項有:

`10pt`, `11pt`, `12pt`, `twoside`, `twocolumn`

等。一般教科書上的字體大小為 `10pt`, 而 `11pt` 為 `10pt` 的 1.1 倍, `12pt` 為 `10pt` 的 1.2 倍。一般論文或一般文件常用 `12pt`。`'twoside'` 是雙面列印, `'twocolumn'` 則為一頁有二欄的格式。

- (d) 在 `\documentclass` 之後, 也可以加入一個或多個 `\usepackage` 的指令, 以表示選用某些 `package`, 例如可以加入:

```
\usepackage{amsmath}           %表示選取 美數學學會  
                                %的數學軟體  
                                %"amsmath.sty"  
\usepackage{amsmath,amssymb} %表示選取 '美數學學會  
                                %的數學軟體  
                                %"amsmath.sty" 及 美數  
                                %學學會數學  
                                %符號軟體 "amssymb.sty"  
\usepackage{colordvi}        %表示選取 "colordvi.sty"
```

- (e) 在 `\usepackage` 之後也可以有：
`\input` 及其他指令
- (f) 接著必須以
`\begin{document}` 表示文件內容本身的開始，
 而以
`\end{document}` 表示文件內容本身的結束。
 因此，LaTeX 不理會 `\end{document}` 以後的所有文字與敘述。

現在再看一個例子—LaTeX 文件檔 `testp2.tex` 如下：

```

%-----
\documentclass[12pt]{article}
\usepackage{amsmath,amssymb}
\usepackage{colordvi}
\begin{document}
\KAI % 楷書之字形指令
\textRed % \textRed 為可使文字等為紅色的指令
        % 須先有 \usepackage{colordvi}
\section{數學與文字敘述}
\textBlue % \textRed 為可使文字等為藍色的指令
  令  $H$  為一 Hilbert 空間,  $C$  為  $H$  之一封閉
  有界的子集合,  $T$  為一  $C$  至本身之 nonexpansive
  映射。假設當  $n \rightarrow \infty$  時, 對每一  $k$ ,
  皆有  $a_{n,k} \rightarrow 0 \cdots$ 
\textBlue
\section{多行數學式子}
\textBlue
\[ % \[ 與 \] 之間者為顯示型數學式子
  F(x)=
  \begin{cases}
    1 & \text{當 } (x > 0) \text{ 時} \\
    0 & \text{當 } (\leq 0) \text{ 時}
  \end{cases} % \begin{cases}... \end{cases}
          % 為條件式環境指令 以後再說明
\]
\end{document}
%-----

```

此文件檔經編譯後，可由預視得知結果為如後：

(若你是初學者，你可以嘗試打一篇如上之 LaTeX 文件，你也可以用 Acrobat Reader 的 ”文字選取工具” 選取，複製上面之段落，再貼入你的編輯視窗。如你的編輯視窗背景色為黑色，你會看不到貼入的文字，可在從編輯視窗右上角 含有文字 ”fixedsys” 可拉下方塊，拉下 選取 ”選取文字顏色: Green” 再到編輯視窗，選取剛才貼入的部分，就可看到貼入的部分。)

上面文件檔 testp2.tex 經編譯預視之結果如下:

1 數學與文字敘述

令 H 為一 Hilbert 空間, C 為 H 之一封閉有界的子集合, T 為一 C 至本身之 nonexpansive 映射。假設當 $n \rightarrow \infty$ 時, 對每一 k , 皆有 $a_{n,k} \rightarrow 0 \dots$

2 多行數學式子

$$F(x) = \begin{cases} 1 & \text{當 } x > 0 \text{ 時} \\ 0 & \text{當 } x \leq 0 \text{ 時} \end{cases}$$

5.2 標題頁(Title Page)

一 LaTeX 文件也可以有 標題頁(Title Page), 其上可以有文件之標題、作者、日期等。

標題頁的內容格式形如：

```
\title{簡單數學文件}
\author{R. Jhon \and J. Peter\
        S. Dunford}
\date{5 October 2003}
.....
\maketitle
```

標題頁係由指令 `\maketitle` 所產生。

指令 `\maketitle` 須置於 `\begin{document}` 之後。

`\title`, `\author`, `\date` 則須置於 `\maketitle` 之前。

現在看一個含有標題頁的例子—LaTeX 文件檔 `testp3.tex` 如下：

```
%-----  
\documentclass[12pt]{article}  
\usepackage{amsmath,amssymb}  
\usepackage{color}  
\begin{document}  
  \kaiu % 楷書字形指令  
  \title{\red 簡單數學文件  
    \thanks{本文件利用 \ChiTeX 編寫}}  
    %\thanks 用以產生註腳  
  \vskip -.2in  
  \author{\green 王\維\thanks{數學系}}  
  \date{}  
  \maketitle  
  \KAI  
  \color{red}  
  \section{數學與文字敘述}  
  \color{blue}  
  令  $H$  為一 Hilbert 空間,  $C$  為  $H$  之一封閉  
  有界的子集合,  $T$  為一  $C$  至本身之 nonexpansive  
  映射。假設當  $n \rightarrow \infty$  時, 對每一  $k$ ,  
  皆有  $a_{n,k} \rightarrow 0 \dots$   
\end{document}  
%-----
```

可得：

簡單數學文件*

王維†

1 數學與文字敘述

令 H 為一 Hilbert 空間, C 為 H 之一封閉有界的子集合, T 為一 C 至本身之 nonexpansive 映射。假設當 $n \rightarrow \infty$ 時, 對每一 k , 皆有 $a_{n,k} \rightarrow 0 \cdots$

2 多行數學式子

$$F(x) = \begin{cases} 1 & \text{當 } x > 0 \text{ 時} \\ 0 & \text{當 } x \leq 0 \text{ 時} \end{cases}$$

*本文件利用 χTeX 編寫
† 數學系

6 LaTeX 文件的語法(二)

- (a) 可出現在 LaTeX 文件的字元符號:
大小寫字母, 數字 $0, \dots, 9$ 及標點符號及
 $. : ; , ? ' ' ! () [] - / *$
- (b) 下列十個特殊字元只當做 LaTeX 的指令:
 $\# \$ \% \& \sim _ \^ \backslash \{ \}$
- (c) i. LaTeX 以 space(空白) (連續的多個空白仍視為一空白) 分隔 word (單字) 如:
This is a test.
- ii. LaTeX 以一個或多個空白行來結束一個 "段" (paragraph)。
因此 (連續的多個空白行與一空白行的作用是一樣的), 例如

```

%-----
Describing simple sentences and paragraphs
to \LaTeX poses no problem; you pretty much
type what comes naturally.

Most \LaTeX commands
describe the logical structure of the document.
%-----

```

可以得到:

Describing simple sentences and paragraphs to \LaTeX poses no problem; you pretty much type what comes naturally.
Most \LaTeX commands describe the logical structure of the document.

(d) 如何打出特殊符號

\$ % & ~ _ ^ \ { }

要打出這些特殊符號, 只要下命令:

\# \\$ \% \& \~ _ \^ \\$\backslash\$ \{ \}

(e) 避免換行與強迫換行:

(i) 避免換行: 要 Mr. Jhon 兩個字不會因換行而被拆開,
可下指令: $\text{Mr}\sim\text{Jhon}$ 或 $\text{\mbox}\{\text{Mr. Jhon}\}$ 。

(ii) 強迫換行: 下指令 \newline 或 \
($\text{\}\text{[+1em]}$ 可換行且與下一行之距離多出 1em)

(f) ① { 與 } 用來界定範圍。

② ^ 及 _ 用來產生上下標 例如: x^2 , a_1 可得 x^2 , a_1

③ & 用於陣列、矩陣、表格等。

(g) LaTeX 以 % 當做一行中 '註解'(Remark) 的開始, 因此 LaTeX 不理會 % 及其右邊之所有文字符號與指令。

7 如何打數學式子

LaTeX 中有兩種數學式子:

- (a) 文句行中的數學式子: 出現在指令 `\(` 與 `\)` 之間的 (或出現在指令 `$` 與 `$` 之間的就是 '文句行中的數學式子', 例如:

```
%-----  
文句行中的數學式子: \(\int_0^x f(t)\,dt\)  
  
                    (\int_0^x f(t)\,dt$ )  
%-----
```

可得: 文句行中的數學式子: $F(x) = \int_0^x f(t) dt$

- (b) 顯示型的數學式子: 出現在指令 `\[` 與 `\]` 之間的就是 '顯示型的數學式子', 它會 '顯示' 在單獨一段的中間, 例如:

```
%-----  
\[\frac{a}{b}\quad \frac{x+y}{1+\frac{y}{z+1}}\]  
%及  
\[\sum_{i=1}^n x_i = \int_0^1 f(x)\,dx\]  
%及  
  \[\int \sin\theta\,d\theta = \cos\theta + C\]  
%-----
```

可得:

$$\frac{a}{b} \quad \frac{x+y}{1+\frac{y}{z+1}}$$
$$\sum_{i=1}^n x_i = \int_0^1 f(x) dx$$
$$\int \sin \theta d\theta = \cos \theta + C$$

(TeX/LaTeX 語法說明：

- (i) `\frac` 為分式指令，格式為 `\frac{分子}{分母}`
 - (ii) `\sum` 為可得 \sum 的指令。
 - (iii) ① `\sin` , `\cos` 可得 \sin , \cos ② `\theta` 為希臘字母 θ
③ `\`, 使多一細空白。
-)

8 章節指令

文句組成 '段'(paragraph), 一些段組成 '節'(section), 一些節組成 '章'(chapter), 一些章組成 '部'(part), ...

- (a) 因此 LaTeX 的 article, report 與 book 格式都有下列的 "節類" 指令 (sectioning command):

```
\part      \subsection    \paragraph
\chapter   \subsubsection  \subparagraph
\section
```

下這些命令 LaTeX 會自動產生 部、章、節、段 (或子節, 次子節) 等的標號。

例如：在 article 格式的文件中下指令：

```
\section{簡單數學式子}
  第一 ...
\section{多行數學式子}
  第二 ...
```

可得：

1 簡單數學式子

第一 ...

2 多行數學式子

第二 ...

- (b) LaTeX 的 report 與 book 格式則有章指令 `\chapter`，下這些命令 LaTeX 會自動產生章數標號，chitex 會動產生中文章數標號:
- i. 下或不下指令 `\ctitlename` 之後則文件標題皆為中文, 例如下命令:

```
\ctitlename
\chapter{群論}
```

則可得:

第 1 章

群論

- ii. 下命令 `\cctitlename`，則文件標題皆為中文且數字為中文。例如在其下下命令

```
\cctitlename
\chapter{群論}
```

則可得:

第一章

群論

- iii. 下命令 `\etitlename` 則文件標題皆為英文。
- 例如下命令:

```
\etitlename
\chapter{群論}
```

則可得到:

Chapter 1

群論

9 字形指令

(1) 英文字形指令:

`\rmfamily` %選 Roman 字形 (內定)
`\sffamily` %選 Sans serif 字形
`\ttfamily` %選 Typewriter 字形
`\upshape` %選 直立形 字形 `\itshape` %選 Italic 字形 (斜體字)
`\slshape` %選 Slanted 字形 (傾斜字) `\bfseries` %選 Boldface 字形 (粗體)

(2) 中文字形指令:

`\kaiu` % 標楷體
`\mingu` % 細明體
`\MING` % 細明體 且用 Adobe Type 字形
`\KAI` % 楷書 且用 Adobe Type 字形
`\LI` % 隸書 且用 Adobe Type 字形
`\FS` % 仿宋 且用 Adobe Type 字形
`\YUAN` % 圓體 且用 Adobe Type 字形
`\WUMING` % 吳明體 (cwtex之中明體 Adobe Type字形)
`\WUFS` % 吳仿宋 (cwtex之吳仿宋 Adobe Type字形)
`\HEI` % 黑體 (cwtex之黑體 Adobe Type 字形)
`\MOESUNG` % 教育部宋體 且用 Adobe Type 字形
`\MOEKAI` % 教育部楷書 且用 Adobe Type 字形
`\moesung` % 教育部宋體
`\moekai` % 教育部楷書
`\idming` % 中國龍明體
`\kaiub` % 粗體標楷體 (用於使用 dvipdfm 時)
`\kaiubi` % 粗斜體標楷體 (用於使用 dvipdfm 時)
`\mingub` % 粗體細明體 (用於使用 dvipdfm 時)
`\mingubi` % 粗斜體細明體 (用於使用 dvipdfm 時)
`\moesungb` % 粗體教育部宋體 (用於使用 dvipdfm 時)
`\moesungbi` % 粗斜體教育部宋體 (用於使用 dvipdfm 時)

```

\moekaib % 粗體教育部楷書 (用於使用 dvipdfm 時)
\moekaibi % 粗斜體教育部楷書 (用於使用 dvipdfm 時)
\idmingb % 粗體中國龍明體 (用於使用 dvipdfm 時)
\idmingbi % 粗斜體中國龍明體 (用於使用 dvipdfm 時)

```

下面是使用字形命令的例子：

```

%-----
\kaiu % \kaiu 為標楷體的指令。開始使用標楷體，
楷書 %以下除非另外下中文字型指令，否則
%皆用 "標楷體"
{LI 文字} % 在大括弧中的指令只有大括弧
% 中有效，因此大括弧中的 "文字" 為隸書
中文 % 在大括弧外的下面之 "中文" 仍為 標楷體
\YUAN %開始使用圓體
字形

\bye % 這是一 plain TeX 文件
% plain TeX 文件須以 \bye 結束
%-----

```

(說明：文件中 % 及其右方的皆當做註解用, 編譯時不理會這些)
可得到：

```

楷書
文字
中文
字形

```

(3) 字形放大指令

(i) 有段放大指令;

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

例如: `{\Large font 字形}` 可得:

font 字形

(ii) 無段放大指令 (chitex 指令):

下指令:

```
\magnification 2000
```

可將中英文字形放大 2.0 倍: 例如:

```
{\magnification 2200 字形 2.2 倍}
```

可得:

字形 2.2 倍

回 chitex 簡介, 字型放大指令

(4) 文句置中指令有二: `\centerline`, 例如:

```
\centerline{This is a test. 測試}
```

```
\begin{center} ... \end{center},
```

例如:

```

\begin{center}
{\large
  Gnus of the World \\
  世界之 Gnus
}
\end{center}

```

可得:

Gnus of the World
世界之 Gnus

10 再談數學式子(一)

1. 上下標, 分別可用 \wedge 及 $_$ 得到
2. 分式, 可用指令 $\frac{\dots\text{分子}\dots}{\dots\text{分母}\dots}$
3. 根號, 用 $\sqrt{\dots}$ 如:
 $\sqrt{x+y}$ and $\sqrt[n]{5}$ $\left(\sqrt{x+y}\right)$ and $\left(\sqrt[n]{5}\right)$
4. 等等之點號, 有:
 x_1, \dots, x_n \$ x_1, \dots, x_n \$
 $a + \dots + z$ \$ a + \dots + z \$
 $n = 1$ 時 $a_n = a + 1$ \$n=1\$ 時 \$a_n=a+1\$
 $n = 2$ 時 $a_n = a^2 + 1$ \$n=2\$ 時 \$a_n=a^2+1\$
 \vdots \$\vdots\$

5. 希臘字母(Greek letters):

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>	η	<code>\eta</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>	ξ	<code>\xi</code>
π	<code>\pi</code>	ϖ	<code>\varpi</code>	ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>	υ	<code>\upsilon</code>
ϕ	<code>\phi</code>	φ	<code>\varphi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>
ω	<code>\omega</code>						

表 1: Lowercase Greek letters

Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>
Ξ	<code>\Xi</code>	Π	<code>\Pi</code>	Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>
Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>	Ω	<code>\Omega</code>		

表 2: Uppercase Greek letters

6. Calligraphic uppercase letters

The letters $\mathcal{A}, \dots, \mathcal{Z}$ are available through use of the style changing command `\cal`. This command behaves like the other style changing commands `\em`, `\it`, etc. so its scope must be delimited as with them. Thus we can type

... for the filter \mathcal{F} we have $\varphi(\mathcal{F}) = \mathcal{G}$.

to obtain

for the filter \mathcal{F} we have $\varphi(\mathcal{F}) = \mathcal{G}$.

There is no need to tabulate all the calligraphic letters, since they are all obtained by just a type style changing command. We will just list them so that we can see, for reference purposes, what they all look like. Here they are:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

±	<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>\wedge</code>	○	<code>\bigcirc</code>
†	<code>\dagger</code>	\	<code>\setminus</code>	∏	<code>\amalg</code>	◦	<code>\circ</code>
‡	<code>\ddagger</code>	·	<code>\cdot</code>	∩	<code>\wr</code>	•	<code>\bullet</code>

表 3: Binary Operation Symbols

7. Binary operators

\LaTeX has been taught to recognise binary operators and set the appropriate space either side of one—i.e., it sets the first argument followed by a little space, then the operator followed by the same little space and finally the second argument. Table 3 shows the binary operators that are available via \LaTeX control words (recall that the binary operators $+$, $-$, and $*$ can be typed from the keyboard). Here are some examples of their use:

<i>Type</i>	<i>To produce</i>
<code>\$a+b\$</code>	$a + b$
<code>\$(a+b) \otimes c\$</code>	$(a + b) \otimes c$
<code>\$(a \vee b) \wedge c\$</code>	$(a \vee b) \wedge c$
<code>\$X - (A \cap B) = (X-A) \cup (X-B)\$</code>	$X - (A \cap B) = (X - A) \cup (X - B)$

8. Binary relations

\LaTeX has been taught to recognize the use of binary relations, too. Table 4 shows those available via \LaTeX control words. There are a few that you can obtain directly from the keyboard: $<$, $>$, $=$, and $|$.

To negate a symbol you can precede the control word that gives the symbol by a `\not`. Some symbols come with ready-made negations, which should be used above the `\not`ing method because the slope of

the negating line is just slightly changed to look more pleasing. Thus `\notin` should be used above `\not\in`, and `\neq` should be used above `\not =`.

If negating a symbol produces a slash whose horizontal positioning is not to your liking, then use the math spacing characters described in section ?? to adjust it.

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

表 4: Binary relations

9. Miscellaneous symbols

Table 5 shows a number of general-purpose symbols. Remember that these are only available in maths mode. Note that `\imath` and `\jmath` should be used when you need to accent an *i* or a *j* in maths mode¹—you cannot use `\i` or `\j` that were available in paragraphing mode. To get a prime symbol, you can use `\prime` or you can just type `'` when in maths mode, as in `\f''(x)=x` which produces $f''(x) = x$.

10. Arrow symbols

\LaTeX has a multitude of arrow symbols, which it will set the correct space around. Note that vertical arrows can all be used as delimiters—see section ?. The available symbols are listed in table 6.

11. Expression delimiters

¹See section ??

\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>	\triangle	<code>\triangle</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>	\diamond	<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 5: Miscellaneous symbols

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Longleftarrow	<code>\Longleftarrow</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>	\Longleftrightarrow	<code>\Longleftrightarrow</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>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\leadsto	<code>\leadsto</code>		

表 6: Arrow symbols

14. 顯示型數學式子, 除可用

`\[`

`:`

`\]`

以外, 也可用

`\begin{ equation*}`

`:`

`\end{ equation*}`

或

`\begin{ equation}`

`:`

`\end{ equation}`

11 再談數學式子(二)——陣列、矩陣等

1. Array:

要得到:

$$\begin{array}{cccc} a+b+c & \text{乙} & x-y & 27 \\ \text{甲} & u+v & z & 134 \\ a & u+uv & xyz & 2978 \end{array}$$

只需用下面指令:

```
$
\begin{array}{cccc}
a+b+c & & \mbox{乙} & & x-y & & 27 & \\
\mbox{甲} & & u+v & & z & & 134 & \\
a & & u+uv & & xyz & & 2978 & \\
\end{array}
```

\$

(數學式子中的中文必須用 \mbox '包' 起來)

要得到:

$$\left(\begin{array}{cccc} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 134 \\ a & u+uv & xyz & 2978 \end{array} \right)$$

只需用下面指令:

```
$
\left(
\begin{array}{cccc}
a+b+c & uv & x-y & 27 \\
a+b & u+v & z & 134 \\
a & u+uv & xyz & 2978
\end{array}
\right)
```

\$

要得到:

$$\begin{vmatrix} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 134 \\ a & u+uv & xyz & 2978 \end{vmatrix}$$

只需用下面指令:

```
$
\left|
\begin{array}{clcr}
a+b+c & uv & x-y & 27 \\
a+b & u+v & z & 134 \\
a & u+uv & xyz & 2978
\end{array}
\right|
$
```

要得到:

$$x = \begin{cases} y & \text{if } y > 0 \\ z + y & \text{oterwise} \end{cases}$$

只需用下面指令:

```
\[ x= \left\{ \begin{array}{ll}
y & \mbox{if } \$y>0\$ \\
z+y & \mbox{oterwise}
\end{array} \right.
\]
```


2. The Matrix Environment

(須用 `\usepackage{amsmath}`)

要得到:

$$\begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

只需用下面指令:

```
$  
\begin{matrix}  
  0 & 1 \\  
  1 & 0  
\end{matrix}  
\quad  
\begin{pmatrix}  
  0 & 1 \\  
  1 & 0  
\end{pmatrix}  
\quad  
\begin{bmatrix}  
  0 & 1 \\  
  1 & 0  
\end{bmatrix}  
$
```

要得到: $\begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}$ $\begin{Vmatrix} 0 & 1 \\ 1 & 0 \end{Vmatrix}$

只需用下面指令:

```
$  
\begin{vmatrix}  
  0 & 1 \\  
  1 & 0  
\end{vmatrix}  
\quad  
\begin{Vmatrix}  
  0 & 1 \\  
  1 & 0  
\end{Vmatrix}  
$
```

12 再談數學式子(三)—條件式、多行數學式子等

1. 條件環境

要得到:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \quad (1)$$

只需用下面指令:

```
\begin{equation}  
  f(x) =  
    \begin{cases}  
      0 & \text{if } $x<0$ \\  
      1 & \text{if } $x\ge 0$  
    \end{cases}  
\end{equation}
```

2. 多行數學式子

要得到:

$$\begin{aligned}(a+b)^3 &= (a+b)(z+b)^2 \\ &= (a+b)(a^2+2ab+b^2) \\ &= a^3+3a^2b+3a^2b+b^3\end{aligned}$$

只需用下面指令:

```
\[
\begin{split}
(a+b)^3 &=(a+b)(z+b)^2 \\
&=(a+b)(a^2+2ab+b^2) \\
&=a^3+3a^2b+3a^2b+b^3
\end{split}
\]
```

要得到:

$$\begin{array}{ll}L_1 = R_1 & L_2 = R_2 \\L_3 = R_3 & L_4 = R_4\end{array}$$

只需用下面指令:

```
\begin{alignat}{2}
L_1=R_1 & \quad L_2=R_2 & \nonumber \\
L_3=R_3 & \quad L_4=R_4 & \nonumber
\end{alignat}
```

13 引述、條列、詩文環境指令

1. 引述指令, 有二: 'quote' 與 'quotation'

(1) 簡短引述指令 'quote'

下面是一利用 'quote' 來做簡短的突顯引述的一例子:

重要的事項可用 'quote' 環境來突顯, 也使得它看來更清楚醒目. It's a good idear to make your input file as easy to read as possible.

它在左右兩邊都向內縮. 要得到上面結果, 只要編寫:

```
%-----  
\begin{quote}  
    重要的事項可用 'quote' 環境來突顯, 也使得它看來更  
    清楚醒目.  
    It's a good idear to make your input file as easy to read  
    as possible.  
\end{quote}  
%-----
```

(2) 較長之 '區段' 引述指令 'quotation' 'quotation' 環境用以突顯超過一段之 '區段' 的引述, '段' 也是由空白行來區隔, 下面是個例子:

'quotation' 環境也可用在其它地方。閒來無事, 何不喝杯茶, 再優游尋覓各種未知環境。'quotation' 環境也可用在其它地方。閒來無事, 何不喝杯茶, 再優游尋覓各種未知環境。

要得到上面結果, 只要編寫:

```

%-----
\begin{quotation}
'quotation' 環境也可用在其它地方。閒來無事，何不
喝杯茶，再優游尋覓各種未知環境。
'quotation' 環境也可用在其它地方。閒來無事，何不
喝杯茶，再優游尋覓各種未知環境。
\end{quotation}
%-----

```

2. 條列(Lists)環境有: `itemize`(項目狀), `enumerate`(計數) 與 `description`(描述)

- 這三種環境的每一新條列項目都以 `\item` 為開始
- 條列(Lists)至少需二個項目.

(1) `itemize`(項目狀) 的環境: `itemize` 的每一項目都以一小實圓圈為開始標記, 例如:

```

\begin{itemize}
\item The first item.
\item The Second item.
\item The Third item.
\end{itemize}

```

會產生:

- The first item.
- The Second item.
- The Third item.

(2) `emumerate` 環境: 在 `enumerate` 的條列環境中的項目是以數字或字母為計數標記, 會依出現次序自動賦予對應的數字或字母. 例如:

```

%-----
\begin{enumerate}
\item 計值:
      \[ \int_0^{\infty} \frac{1}{1+x^2} dx \]
\item 解出  $x$  :
      \[
\begin{vmatrix}
x-1 & 1 & 0 \\
2 & x & 1 \\
1 & x & 1
\end{vmatrix}
= 0
\]
\end{enumerate}
%-----

```

可以產生:

1. 計值:

$$\int_0^{\infty} \frac{1}{1+x^2} dx$$

2. 解出 x :

$$\begin{vmatrix} x-1 & 1 & 0 \\ 2 & x & 1 \\ 1 & x & 1 \end{vmatrix} = 0$$

- 條列(Lists) 可以是巢狀地在每一條列中又有條列, 例如:

```

%-----
\begin{enumerate}
\item 求值:
  \begin{enumerate}
    \item[(1)]  $\lim_{x \rightarrow \infty} x^{\sin x}$ 
    \item[(2)]  $\displaystyle \int_0^{\infty} \frac{1}{1+x^2} dx$ 
  \end{enumerate}
\item 微分:  $x^{\sin x}$ 
\end{enumerate}
%-----

```

可以產生:

1. 求值:

(1) $\lim_{x \rightarrow \infty} x^{\sin x}$

(2) $\int_0^{\infty} \frac{1}{1+x^2} dx$

2. 微分: $x^{\sin x}$

- 項目 `\item` 可有選項可以自定標號, 下命令 `\item[text]`, 則項目之標號為 `text`, 例如:

```

%-----
\item[(1)] The first sentence.
\item[(2)] The second sentence.
%-----

```

可以產生:

-
- (1) The first sentence.
(2) The second sentence.
-

(3) description 環境:

```
%-----  
  \begin{description}  
    \item[杜甫] The first sentence.  
    \item[蘇東坡] The second sentence.  
  \end{description}  
%-----
```

可以產生:

杜甫 The first sentence.

蘇東坡 The second sentence.

3. 'verse' 詩篇環境:

```
%-----  
\begin{verse}  
  There is an environment for verse \\  
  Whose features some poets will curse.  
  For instead of making \\  
  Them do {\em all\/} line breaking, \\  
  It allows them to put too many words  
\begin{verse}  
  on a line when they'd rather be\\  
  forced to be terse.  
\end{verse}  
\end{verse}  
%-----
```

可以產生:

There is an environment for verse
Whose features some poets will curse. For instead of making
Them do *all* line breaking,
It allows them to put too many words
on a line when they'd rather be
forced to be terse.

14 行距指令

用 `\baselineskip` 可調整行距, 通常單間距 (single space) 為 12pt, 即, 一

般內定為 `\baselineskip=12pt` 若要雙間距可下命令 `\baselineskip=24pt`
當然也可設為其它值, 例如 `\baselineskip=17pt`