

L^AT_EX III:

基本排版、修飾、自訂指令

蔡炎龍, 政治大學應用數學系

2013 年 8 月於國立清華大學

- 1 L^AT_EX 的排版
- 2 L^AT_EX 的基本修飾
- 3 自訂指令
- 4 在 L^AT_EX 打入程式碼
- 5 目錄和章節排版
- 6 進階中文 L^AT_EX 技巧
- 7 學習資訊

L^AT_EX 的排版

Geometry 套件的基本使用法

用 `geometry` 套件可以很容易安排版面規畫。我們在預設區引入 `Geometry` 的方式, 可以是:

```
\usepackage[你要下的參數]{geometry}
```

或是

```
\usepackage{geometry}
```

```
\geometry{你要下的參數}
```

Geometry 套件的基本使用法

比方說, 下列兩種方式是一樣的:

方法一:

```
\usepackage[a4paper,margin=1in]{geometry}
```

方法二:

```
\usepackage{geometry}
```

```
\geometry{a4paper,margin=1in}
```

紙張及邊界設定

我們先來看 `geometry` 紙張及邊界設定的參數：

<code>a4paper</code>	指定使用 A4 紙張
<code>margin=1in</code>	把四邊邊界都設成 1 英吋
<code>left=1in</code>	把左邊邊界設成 1 英吋
<code>right=1in</code>	把右邊邊界設成 1 英吋
<code>hmargin=1in</code>	把左右邊界都設成 1 英吋
<code>top=1in</code>	把上方邊界設成 1 英吋
<code>bottom=1in</code>	把下方邊界設成 1 英吋
<code>vmargin=1in</code>	把上下方邊界都設成 1 英吋

geometry 其他常用參數

`textwidth=17cm`

內文寬度設為 17 cm

`textheight=25cm`

內文高度設為 25 cm

`bindingoffset=0.5in`

奇數頁左邊，偶數頁的右邊，會
多出 0.5 英吋的裝訂空間

`nohead`

去掉頁首

`nofoot`

去掉頁尾的 `footskip`

段落間距、行距

我們介紹 L^AT_EX 段落間距、行距等設定方式：

`\linespread{1.2}` 行距設為原本的 1.2 倍

段落間距、行距

我們介紹 L^AT_EX 段落間距、行距等設定方式：

`\linespread{1.2}` 行距設為原本的 1.2 倍

`\setlength{\parskip}{15pt}` 段落間距設為 15 點

段落間距、行距

我們介紹 L^AT_EX 段落間距、行距等設定方式：

`\linespread{1.2}` 行距設為原本的 1.2 倍

`\setlength{\parskip}{15pt}` 段落間距設為 15 點

`\setlength{\parindent}{0cm}` 每段首行縮排設為 0

多欄排版

使用 `multicol` 套件, 我們可以進行多欄排版。比方說, 在開始要用兩欄的地方, 我們可以打入:

```
\begin{multicols}{2}
```

```
:
```

```
:
```

```
\end{multicols}
```

L^AT_EX 的基本修飾

L^AT_EX 的常用長度單位

我們已經偷偷下了好幾次 L^AT_EX 長度指令, 但倒底 L^AT_EX 的長度單位有哪些呢? 這裡把它們列出來:

cm 公分

mm 公厘

in 公吋

pt 點

pc pica, 12 點

L^AT_EX 的常用長度單位

我們已經偷偷下了好幾次 L^AT_EX 長度指令, 但倒底 L^AT_EX 的長度單位有哪些呢? 這裡把它們列出來:

cm 公分

mm 公厘

in 公吋

pt 點

pc pica, 12 點

em 大約為正在使用字型大寫 **M** 字母之寬度

L^AT_EX 的常用長度單位

我們已經偷偷下了好幾次 L^AT_EX 長度指令, 但倒底 L^AT_EX 的長度單位有哪些呢? 這裡把它們列出來:

cm 公分

mm 公厘

in 公吋

pt 點

pc pica, 12 點

em 大約為正在使用字型大寫 M 字母之寬度

ex 大約為正在使用字型小寫 x 字母之高度

L^AT_EX 的空白

`\,` 小空白

L^AT_EX 的空白

`\,` 小空白

`\:` 中空白

L^AT_EX 的空白

`\,` 小空白

`\:` 中空白

`\;` 大空白

L^AT_EX 的空白

`\,` 小空白

`\:` 中空白

`\;` 大空白

`\quad` 更大的空白 (一個 *em*)

L^AT_EX 的空白

`\,` 小空白

`\:` 中空白

`\;` 大空白

`\quad` 更大的空白 (一個 *em*)

`\qquad` 再大的空白 (兩個 *em*)

L^AT_EX 的空白

`\,` 小空白

`\:` 中空白

`\;` 大空白

`\quad` 更大的空白 (一個 *em*)

`\qquad` 再大的空白 (兩個 *em*)

`\hspace{2in}` 乾脆自己設啦

L^AT_EX 的空白

`\,` 小空白

`\:` 中空白

`\;` 大空白

`\quad` 更大的空白 (一個 *em*)

`\qquad` 再大的空白 (兩個 *em*)

`\hspace{2in}` 乾脆自己設啦

L^AT_EX 的空白

`\,` 小空白

`\:` 中空白

`\;` 大空白

`\quad` 更大的空白 (一個 *em*)

`\qquad` 再大的空白 (兩個 *em*)

`\hspace{2in}` 乾脆自己設啦

垂直方向的空白

`\smallskip` 略往下移一點點 (約 3 點)

垂直方向的空白

`\smallskip` 略往下移一點點 (約 3 點)

`\medskip` 再大一點點 (約 6 點)

垂直方向的空白

`\smallskip` 略往下移一點點 (約 3 點)

`\medskip` 再大一點點 (約 6 點)

`\bigskip` 更大一點 (約 12 點)

垂直方向的空白

`\smallskip` 略往下移一點點 (約 3 點)

`\medskip` 再大一點點 (約 6 點)

`\bigskip` 更大一點 (約 12 點)

`\vspace{2in}` 愛空多少空多少

`\hfill` 和 `\vfill`

`\hfill` 和 `\vfill` 是很有意思的兩個指令，作用是「很平均的把水平 (或垂直) 空間塞進去」，能塞多少就塞多少。

\hfill 和 \vfill

\hfill 和 \vfill 是很有意思的兩個指令,作用是「很平均的把水平(或垂直)空間塞進去」,能塞多少就塞多少。請試驗下面的 L^AT_EX 碼:

姓名: 蔡炎龍 \hfill 學號: 123 \hfill 日期: 05/09/08

\hfill 和 \vfill

\hfill 和 \vfill 是很有意思的兩個指令,作用是「很平均的把水平(或垂直)空間塞進去」,能塞多少就塞多少。請試驗下面的 L^AT_EX 碼:

姓名: 蔡炎龍 \hfill 學號: 123 \hfill 日期: 05/09/08

另外,還有兩個和 \hfill 類似,只是不是塞空白的指令:

\dotfill 塞點進去

\hrulefill 塞一條線進去

字型的大小

`\tiny` 超小

字型的大小

`\tiny` 超小

`\small` 小

字型的大小

`\tiny` 超小

`\small` 小

`\large` 大

字型的大小

`\tiny` 超小

`\small` 小

`\large` 大

`\Large` 再大

字型的大小

`\tiny` 超小

`\small` 小

`\large` 大

`\Large` 再大

`\LARGE` 再再大

字型的大小

`\tiny` 超小

`\small` 小

`\large` 大

`\Large` 再大

`\LARGE` 再再大

`\huge` 極大

字型的大小

`\tiny` 超小

`\small` 小

`\large` 大

`\Large` 再大

`\LARGE` 再再大

`\huge` 極大

`\Huge` 超級大

字型的大小

`\tiny` 超小

`\small` 小

`\large` 大

`\Large` 再大

`\LARGE` 再再大

`\huge` 極大

`\Huge` 超級大

`\HUGE` 沒這個指令啦, 前面已經最大了!

強調

L^AT_EX 的標準強調用法有：

`\emph{words}` 通常是斜體，不過是依 `class` 內的設定而可能不同

強調

L^AT_EX 的標準強調用法有：

`\emph{words}` 通常是斜體，不過是依 `class` 內的設定而可能不同

`{\bf{words}}` 粗體

強調

L^AT_EX 的標準強調用法有：

`\emph{words}` 通常是斜體，不過是依 `class` 內的設定而可能不同

`{\bf{words}}` 粗體

強調

L^AT_EX 的標準強調用法有：

`\emph{words}` 通常是斜體，不過是依 `class` 內的設定而可能不同

`{\bf{words}}` 粗體

使用中文時，有時字型沒有「真正」的粗體，有時粗體也不特別明顯，此時可以用其他字型來做強調。

定理的字型

定要的顯示的「定理」字型, 在設定時改變即可。例如

```
\newtheorem{thm}{\fbox{定理}}
```

頁首頁尾設定

使用 `fancyhdr` 可以自由設定頁首頁尾。

```
\pagestyle{fancy}
```

```
\thead{}
```

```
\chead{}
```

```
\rhead{}
```

```
\tfoot{}
```

```
\cfoot{}
```

```
\rfoot{}
```

```
\renewcommand{\headrulewidth}{2pt}
```

```
\renewcommand{\footrulewidth}{2pt}
```

特殊符號打法

“引號” `` 和 "

特殊符號打法

“引號” `` 和 "

`%, {, }, &, $, # \%, \{, \}, \&, \$, \#`

特殊符號打法

“引號” `` 和 "

`%, {, }, &, $, # \%, \{, \}, \&, \$, \#`

`\ \backslash 或 \textbackslash$`

特殊符號打法

“引號” `` 和 "

`%, {, }, &, $, # \%, \{, \}, \&, \$, \#`

`\ \backslashslash 或 \textbackslashslash`

`ä \"a`

特殊符號打法

“引號” `` 和 "

`%`, `{`, `}`, `&`, `$`, `#` `\%`, `\{`, `\}`, `\&`, `\$`, `\#`

`\` `\$``\backslash``slash` 或 `\textbackslash`

`ä` `\"a`

`è`, `é` `\`e`, `\'e`

自訂指令

基本的自訂指令方式

使用 L^AT_EX, 不久之後, 你就會發現重覆打同一堆指令很麻煩。

基本的自訂指令方式

使用 L^AT_EX, 不久之後, 你就會發現重覆打同一堆指令很麻煩。比方說, 你可能常用 \mathbb{R} 表示實數 \mathbb{R} , 但每次都要打這麼長的指令有點辛苦。

基本的自訂指令方式

使用 L^AT_EX, 不久之後, 你就會發現重覆打同一堆指令很麻煩。比方說, 你可能常用 \mathbb{R} 表示實數 \mathbb{R} , 但每次都要打這麼長的指令有點辛苦。

如果, 你希望以後用 `\R` 來代表 `\mathbb{R}`, 可以在預設區用 `\newcommand` 來做:

```
\newcommand{\R}{\mathbb{R}}
```

從此, 你只要下達 `\R`, 就和打入 `\mathbb{R}` 是一樣的!

向量的例子

假設我們想做一個像這樣的向量表示：



可以用

```
\overset{\rightharpoonup}{A}
```

向量的例子

我們希望自訂一個向量指令 `\vect`, 以後打入 `\vect{A}`, 就出現 \vec{A} 。這裡我們可以在全文設定區做一個新的指令:

```
\newcommand{\vect}{\overset{\rightharpoonup{}}
```

帶參數的指令

有時, 我們選用明體為主要字體, 但不時可能要用楷書等其他字體。假設我們用 `xeCJK` 設了

```
\setCJKfamilyfont{kaiti}{標楷體} % 或其他楷書字型  
{\CJKfamily}{kaiti} 我是楷書
```

帶參數的指令

我想很少人會喜歡每次這樣打, 但我們又不能直接用前面的方式, 因為我們想用的內容不一定是“我是楷書”。這時, 我們可以用指定需要加入參數的方式:

```
\newcommand{\kai}[1]{\CJKfamily{kaiti} #1}
```

帶參數的指令

我想很少人會喜歡每次這樣打, 但我們又不能直接用前面的方式, 因為我們想用的內容不一定是“我是楷書”。這時, 我們可以用指定需要加入參數的方式:

```
\newcommand{\kai}[1]{\CJKfamily{kaiti} #1}
```

以後, 你只要打入

```
\kai{我是楷書}
```

就能得到相同的結果。

指令和 L^AT_EX 預設相同

今天, 你本來想用 `\v` 來代表 `\mathbf{v}`, 就是 \mathbf{v} 這根向量。於是你設了

```
\newcommand{\v}{\mathbf{v}}
```

結果, L^AT_EX 會告訴你:

```
LaTeX Error: Command \v already defined.
```

指令和 L^AT_EX 預設相同

今天, 你本來想用 `\v` 來代表 `\mathbf{v}`, 就是 **v** 這根向量。於是你設了

```
\newcommand{\v}{\mathbf{v}}
```

結果, L^AT_EX 會告訴你:

LaTeX Error: Command \v already defined.

原來, 在 L^AT_EX 裡, 如果你打入 `\v{c}` 會出現 č。

指令和 L^AT_EX 預設相同

今天, 你本來想用 `\v` 來代表 `\mathbf{v}`, 就是 \mathbf{v} 這根向量。於是你設了

```
\newcommand{\v}{\mathbf{v}}
```

結果, L^AT_EX 會告訴你:

LaTeX Error: Command \v already defined.

原來, 在 L^AT_EX 裡, 如果你打入 `\v{c}` 會出現 \check{c} 。如果你覺得對你來說並不重要, 可以改變打入 `\v` 就出現粗體的 \mathbf{v} , 那可以用 `\renewcommand`:

```
\renewcommand{\v}{\mathbf{v}}
```

新的數學函數

我們注意看 L^AT_EX 內建數學函數都是正體字，變數等用斜體。要是 L^AT_EX 沒有定的，比如說矩陣的 `rank`，我們這樣打：

```
 $\$rank A\$$ 
```

會出現 *rankA*，真是非常醜！我們應該在 **全文設定區** 加入新的數學函數：

```
\DeclareMathOperator{\rank}{rank}
```

現在打出 $\$ \backslash rank A \$$ 會出現 **rank A**!

在 L^AT_EX 打入程式碼

隨文程式碼

請問要在 L^AT_EX 中打出 `\LaTeX` 要怎麼做呢？我們來介紹一個有趣的指令叫 `\verb`。剛剛的問題可以這樣解決：

`\verb|\LaTeX|`

所以我們要用 “|” 把我們想打出的括起來嗎？其實是隨便。`\verb` 指令有趣的是它會把第一個讀到的當分隔。比方說下列打法也是一樣結果：

`\verb^\LaTeX^`

多行程式碼

「一般」我們要打程式碼可以用 `verbatim` 環境, 比如說我們來一小段簡單 Python 程式:

```
\begin{verbatim}  
for i in range(10):  
    print(i)  
\end{verbatim}
```

結果

```
for i in range(10):  
    print(i)
```

多行程式碼

有一個更好用的套件叫 **minted**。

```
\begin{minted}{python}
for i in range(10):
    print(i)
\end{minted}
```

結果

```
for i in range(10):
    print(i)
```

多行程式碼

不過 `minted` 套件需要先裝 `Python` 程式語言, 然後安裝一個叫 `Pygments` 的程式。基本上高亮化的工作是 `Pygments` 做的。

然後執行時要允許 `LATEX` 執行外部程式, 要加入 `shell-escape` 參數, 例如:

```
xelatex --shell-escape foo.tex
```

詳情請搜尋「炎龍老師 `minted`」:)

目錄和章節排版

目錄

要在 L^AT_EX 做目錄其實非常容易, 就在文章開始的地方, 下達:

`\tableofcontents`

目錄的深度

所謂目錄的深度就是你準備把第幾層的結構都放進目錄中。如果說你想更改, 例如到第 4 層 paragraph 都要包進來, 就是在全文設定區下達:

```
\setcounter{tocdepth}{4}
```

進階中文 L^AT_EX 技巧

來寫中文的一二三

我們來試試在使用 `xeCJK` 時, 加入一個參數:

```
\usepackage[CJKnumber]{xeCJK}
```

這有什麼用呢? 我們可以在內文中, 試試:

```
\CJKnumber{3}
```

編譯後真的出現「三」! 你還可以試試 `\CJKnumber{53}` 看看, 很有趣的。

中文項次

我們想要用 `enumerate` 環境得到：

結果

- 一、微積分
- 二、高得微積分
- 三、線性代數

這要怎麼做呢？還記得我們可以引入

`\usepackage{enumerate}`，方便客制化項次。不過我們用當然不能用 `\begin{enumerate}[一、]`，因為 L^AT_EX 不懂中文...

cwTeX Q 五套字型

cwTeX Q Kai Medium 楷書

cwTeX Q Hei Bold 黑體

cwTeX Q Ming Medium 明體

cwTeX Q Yuan Medium 圓體

cwTeX Q Fangsong Medium 仿宋體

xeCJK 中文字型的完整設定

我們在設主要字型, 其實還可設對應的粗體、斜體。例如我們設明體為主要字型, 粗體用黑體字, 斜體用仿宋體。

```
\setCJKmainfont[BoldFont={cwTeX Q Hei Bold},  
ItalicFont={cwTeX Q Fangsong Medium}]  
{cwTeX Q Ming Medium}
```

設定 cwT_EX 五套字型

% 楷書

```
\setCJKfamilyfont{kai}{cwTeX Q Kai Medium}
```

% 黑體

```
\setCJKfamilyfont{hei}{cwTeX Q Hei Bold}
```

% 明體

```
\setCJKfamilyfont{ming}{cwTeX Q Ming Medium}
```

% 圓體

```
\setCJKfamilyfont{yuan}{cwTeX Q Yuan Medium}
```

% 仿宋體

```
\setCJKfamilyfont{fsong}{cwTeX Q Fangsong Medium}
```

設定 cwT_EX 五套字型

% 用 \kai{使用楷書}

```
\newcommand{\kai}[1]{\CJKfamily{kai}#1}
```

% 用 \hei{使用黑體}

```
\newcommand{\hei}[1]{\CJKfamily{hei}#1}
```

% 用 \ming{使用明體}

```
\newcommand{\ming}[1]{\CJKfamily{ming}#1}
```

% 用 \yuan{使用圓體}

```
\newcommand{\yuan}[1]{\CJKfamily{yuan}#1}
```

% 用 \fsong{使用仿宋體}

```
\newcommand{\fsong}[1]{\CJKfamily{fsong}#1}
```

設定 XeCJK 英文字型

XeL^AT_EX 方便的地方是中英文可以做不同的設定。

% 指定英文字型, 並用 LaTeX 引號

```
\setromanfont[Mapping=tex-text]{Times}
```

% 指定無描邊字型

```
\setsansfont[Scale=MatchLowercase,Mapping=tex-text]{A
```

% 指定等寬字型

```
\setmonofont[Scale=MatchLowercase]{Courier}
```

中文項次

L^AT_EX 裡有一些計數器 (counter), 比如說紀錄第幾章、第幾節。而紀錄現在是 `enumerate` 第幾項的計數器叫 `enumi`。計數器要用阿拉伯數字顯示的話是用 `\arabic{enumi}`, 然後要再把阿拉伯數字換成中文就是...因此, 我們設 `enumerate` 環境要這樣做:

```
\begin{enumerate}[\CJKnumber{\arabic{enumi}}、]
```

更改 L^AT_EX 預設名稱

我們在用 `figure` 環境時，會出現 `figure 1`, `figure 2` 等字眼，但是我們當然希望出現的是 **圖 1**, **圖 2** 這樣。這要如何做呢？其實很容易，只要改變 L^AT_EX 內設名稱就好。而 `figure` 的內設名稱是放在 `\figurename`，因此我們在全文設定區更改：

```
\renewcommand{\figurename}{圖}
```

更改 L^AT_EX 預設名稱

另外常用到的幾個名稱, 可照樣修改。

<code>\abstractname</code>	摘要
<code>\appendixname</code>	附錄
<code>\bibname</code>	參考文獻 (report, book)
<code>\contentsname</code>	目錄
<code>\listfigurename</code>	圖目錄
<code>\listtablename</code>	表格目錄
<code>\refname</code>	參考文獻 (article)

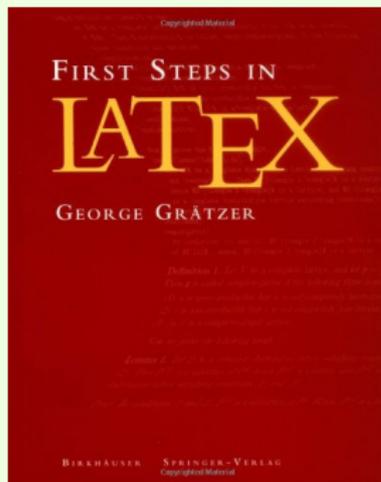
建議套件

titletoc 客制化目錄

titlesec 客制化章節格式

學習資訊

推薦入門書



Frist Steps in L^AT_EX

<http://amzn.com/0817641327>

如果你只準備買一本 L^AT_EX 的書, 就想把論文打出來, 那就是 *George Grätzer* 的這一本。又小本、又把該注意的都提到。可以說是「論文從無到有」最佳入門書。

推薦套件介紹

continued from previous page

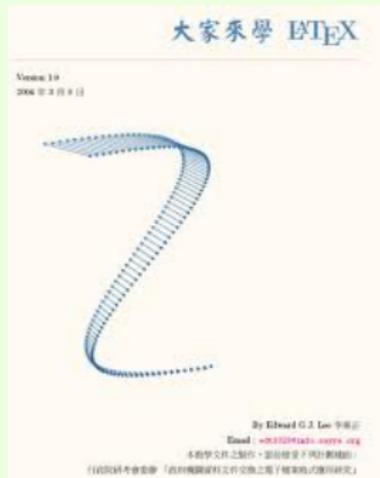
ひらがな	カタカナ	漢字
あ	ア	亜
あ	ア	亞
あ	ア	亜
あ	ア	亞
あ	ア	亜
あ	ア	亞
あ	ア	亜
あ	ア	亞
あ	ア	亜
あ	ア	亞
あ	ア	亜
あ	ア	亞
あ	ア	亜
あ	ア	亞
あ	ア	亜

熊澤教授的 L^AT_EX 套件網

<http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/texindex.html>

這是日本熊澤吉起教授的 L^AT_EX 套件介紹網頁, 幾乎所有常見的套件都有介紹, 而且圖示清楚, 所以不懂日文也能立刻瞭解!

推薦中文入門

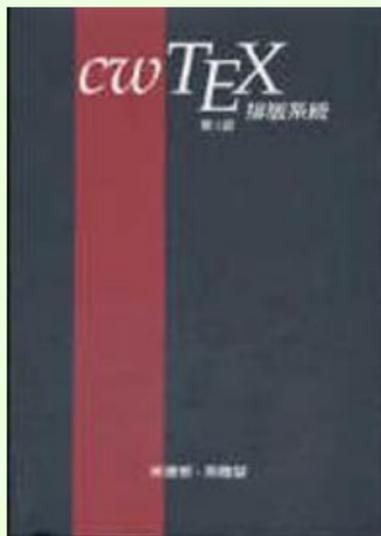


大學來學 L^AT_EX

<http://edt1023.sayya.org/tex/latex123/latex123.html>

這是李果正先生寫的中文 L^AT_EX 入門文件。許多我們沒有說明的細節都有更進一步的說明。

推薦中文深度學習



cwTeX3 手冊

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

這是 cwTeX 的手冊。不過大部份的概念和技巧都是一般 L^AT_EX、中文 L^AT_EX 通用的。這本手冊大概是中文 L^AT_EX 最深入討論各種排版細節問題的書，很值得想深入研討的人仔細研讀。

推薦主題學習



L^AT_EX 的 Wikibook

<http://en.wikibooks.org/wiki/LaTeX>

L^AT_EX Wikibook 裡大概你想知道的主題，比方說表格啦、排版啦都有相當好的介紹。可以不時看看，增強功力！