

ℒ<sub>A</sub>T<sub>E</sub>X 論文樣板與簡易教學

ℒ<sub>A</sub>T<sub>E</sub>X Thesis Template and Tutorial

主要作者：sppmg

中 華 民 國 一 百 零 五 年 十 月



# 前言

這是一個簡易的  $\text{\LaTeX}$  教學。概略的介紹  $\text{\LaTeX}$  的用法，並且使用 Cookbook 的形式寫作。也就是說使用者可以直接複製/貼上  $\text{\LaTeX}$  code，省去慢慢打、慢慢學的時間。除了  $\text{\LaTeX}$  語法外，本文也會介紹一些幫助撰寫  $\text{\LaTeX}$  的工具。所有知識都不會說的太詳細，各位要深入了解的話就請各位自行發揮舉一反三的能力上 Google 搜尋囉！這份文件（以及論文樣板）歡迎各位自由使用、修改、發佈。發現內容有誤也歡迎指正。如果研究生認為我幫你節省很多時間，想感謝一下的話，請推薦更多人閱讀、採用，這樣就可以稀釋我的時間成本了 QQ。



# 目錄

	頁次
前言 .....	iii
目錄 .....	v
圖目錄 .....	ix
表目錄 .....	xi
一、 FAQ .....	1
1.1 什麼是 L <sup>A</sup> T <sub>E</sub> X? .....	1
1.2 這樣板和中央大學 (羅吉昌先生) 提供的有何不同? .....	1
1.3 這樣板符合學校要求嗎? .....	2
1.4 這樣板有很厲害嗎? .....	2
1.5 要怎麼使用樣板? .....	2
1.6 樣板不符合我的要求怎辦? .....	4
1.7 有其他教學嗎? .....	4
1.8 MATLAB 圖要如何插入? .....	4
1.9 編譯出錯怎辦? .....	4
二、 軟體安裝 .....	5
2.1 TeX 編輯器 .....	5
2.2 TeX 發行版 .....	6
2.2.1 GNU/Linux .....	6
2.2.2 Windows .....	7
2.3 文獻資料庫 (.bib) 管理軟體 .....	7

2.4	本 L <sup>A</sup> T <sub>E</sub> X 專案結構及使用說明 .....	9
2.4.1	取得專案 .....	9
2.4.2	專案結構 .....	10
三、	L <sup>A</sup> T <sub>E</sub> X 基礎教學 .....	11
3.1	背景知識 .....	11
3.1.1	T <sub>E</sub> X、L <sup>A</sup> T <sub>E</sub> X 歷史 .....	11
3.1.2	編譯/生成 PDF 檔 .....	11
3.2	文件結構 .....	13
3.3	套件與指令 .....	14
3.4	文字 .....	15
3.4.1	保留字 .....	16
3.4.2	字體大小 .....	16
3.4.3	字型樣式 .....	17
3.5	圖片 .....	18
3.5.1	插入單一圖片 .....	18
3.5.2	插入多張圖片 .....	20
3.6	表格 .....	22
3.6.1	一般表格 .....	22
3.6.2	自動折行表格 .....	22
3.6.3	複雜表格 .....	23
3.6.4	其他表格 .....	23
3.7	列表/條列 .....	23
3.8	數學符號、方程 .....	24
3.8.1	顯示模式 .....	24
3.8.2	常用數學符號 .....	25
3.9	科學符號 .....	26
3.9.1	SI 單位 .....	26
3.9.2	化學式 .....	26

3.10	標記與參考 Cross-referencing.....	26
3.10.1	標記 (labal).....	27
3.10.2	參考 (ref/cref).....	27
3.11	文獻.....	27
3.11.1	嵌入式文獻資料庫.....	28
3.11.2	BibTeX .....	28
3.11.3	BibLaTeX .....	29
3.12	程式碼.....	30
3.13	註解.....	30
3.13.1	L <sup>A</sup> T <sub>E</sub> X 原生註解 .....	30
3.13.2	TodoNotes 套件.....	31
四、	進一步了解 .....	33
4.1	分割文件 .....	33
4.2	字型.....	33
4.3	自製指令與巨集 .....	34
4.4	文件類型 (documentclass) 選項 .....	36
4.5	tex 檔編碼 .....	36
4.6	樣板提供工具 .....	36
4.6.1	圓圈文字 .....	36
4.6.2	快速插入單一圖片.....	37
4.6.3	單位 .....	37
五、	寫作輔助工具 .....	39
5.1	SyncTeX .....	39
5.1.1	Kile + Okuar.....	39
5.2	Git 版本管理與 tex 檔比較 .....	39
5.2.1	gitignore.....	40
六、	文獻搜尋 .....	43

七、	特定樣板細說 .....	45
7.1	中央大學 (NCU) .....	45
7.1.1	章節層級 .....	45
八、	樣板測試結果及已知問題.....	51
8.1	測試結果 .....	51
8.2	待解決問題 .....	51



# 圖目錄

	頁次
2.1 BBT 拖曳插入 cite_key 的設定方法。 . . . . .	9
2.2 . . . . .	9
2.3 GitHub 的分支及標籤列表。 . . . . .	10
3.1 Linux logo . . . . .	18
3.2 . . . . .	20
3.3 line 2, caption . . . . .	20
3.4 . . . . .	20
3.5 line 4, caption . . . . .	20
3.6 line 5, caption . . . . .	20
3.7 line 6, caption . . . . .	20
3.8 short caption . . . . .	20
3.9 caption, 這裡可使用 (b) 取得子圖 (Debian) 編號 . . . . .	21
3.10 Linux logo . . . . .	27
6.1 . . . . .	44
7.1 樣板提供的目錄風格 . . . . .	46
7.2 \styletitles{0} . . . . .	47
7.3 \styletitles{1} . . . . .	48
7.4 \styletitles{1} 與\leftskip . . . . .	49



# 表目錄

	頁次
3.1 指令範例 . . . . .	15
3.2 字體大小範例 . . . . .	17
3.3 樣式範例 . . . . .	17
3.4 Solution . . . . .	22
3.5 常用數學符號 . . . . .	25
3.6 常用標記類型 . . . . .	27



# 一、FAQ

## 1.1 什麼是 L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X 是一個非常古老、強大、專業的排版軟體。它將許多專業排版的細節隱藏起來<sup>1</sup>，讓你專注在內容上而不用擔心最終文件是否好看。這一個特性又以數學公式最為顯著（用 Word 的應該可以深切體會微軟對於防止客戶腦袋老化的用心良苦:D）。

使用上來說，L<sup>A</sup>T<sub>E</sub>X 使用過程如同寫程式般，須要用一些指令告訴 L<sup>A</sup>T<sub>E</sub>X 這段文字是什麼（標題、章節.....）。最後還要經過編譯才能看到排版結果<sup>2</sup>。這種非「所見即所得」(WYSIWYG) 的軟體可能會讓許多新手敬而遠之。不過不用擔心，你已經拿到這份教學了，一定可以在 1hr 內學會寫 L<sup>A</sup>T<sub>E</sub>X 文件的:D！

## 1.2 這樣板和中央大學 (羅吉昌先生) 提供的有何不同?

嗯.... 這對我來說是能否編譯的差別 XD。原因我也不知道為何，不過不少同學即使能夠產生 PDF 檔，也還是須要修改一些小地方。所以我乾脆就把我從頭打造的設定做成樣板啦！<sup>3</sup> 我寫論文時選定的編譯器是 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X，理由是他可以直接使用系統內的字型，也對非拉丁文字支援較好的樣子。所以這個樣板完全只測試 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 下的使用情況。

除了專注在 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 之外，我的版本也儘量使用套件處理格式而不直接動底層的設定。譬如版面設定直接使用 geometry 套件設定邊距。這麼做除了我本身不熟 L<sup>A</sup>T<sub>E</sub>X 外，也相當於將底層的轉換交給套件開發人員，會

---

<sup>1</sup>專業排版連0.2mm都在苛求。

<sup>2</sup>雖然部份編輯器提供即時編譯功能，只要停止輸入幾秒立刻進行編譯。但實務上這不是必要的，並且會加重電腦的負擔。另外有些如 Lyx 這類「所見即所得」(WYSIWYG) 的軟體是基於 L<sup>A</sup>T<sub>E</sub>X，各位可以玩玩看。

<sup>3</sup>其中還是有很小一部份是修改自羅吉昌的樣板，譬如封面頁的格式幾乎就是羅吉昌的。

比我自己維護來的好得多。我也替換掉將數字轉成中文 ( $1 \rightarrow \text{一}$ 、 $2 \rightarrow \text{二}$ ) 的 `xCJKnumb` 而改用 `zhnumber`。這是因為 `xCJKnumb` 為 `CJKnumb` 的改造版，而這個套件不存在於 `ctan` 網站上，要取得只能 google，然後在不知名 blog 上下載。雖然目前使用上沒有問題，但未來維護則未知。

### 1.3 這樣板符合學校要求嗎？

先說結論，我的論文是照此格式提交的，既然審過了，應該沒有問題。

4

這樣板並非完全照學校要求製作。理由是有些規定我認為過於於愚蠢或是使編排變得混亂。差異處如下：

- 附錄編號採用英文字母 A,B,C。理由：附錄編號與正文不該相同。（羅先生樣板雖然表面符合學校格式，但實在不應該讓章條目是中文，節條目出現英文 A.1。）
- 節編號的連接符「-」換成「.」。理由：1.  $\text{\LaTeX}$  預設。2. 表達較為清晰、美觀。3. 沒看過用「-」的。
- 圖、表、公式全部以章分別編號。理由：1.  $\text{\LaTeX}$  預設。2. 這樣做更能了解內容類別。

另外還有章節風格的問題，請見章節 7.1.1。

### 1.4 這樣板有很厲害嗎？

我前後大概花了 1 個月左右來調樣板吧！（然後教學又是 1 個多月.....）有很多小細節都已經解決了，也儘量做到自動化設定。當然啦！花費時間跟天資也是有關的，你要說「這東西也要花一個月？」我也認了 orz.....

### 1.5 要怎麼使用樣板？

整個流程如下：

---

<sup>4</sup>其實中央很松啦～～我看過比我更慘的格式。**重點在內容啊！**

1. 照教學檔安裝、設定軟體環境。修改編輯器的預設 LaTeX 編譯器，改成使用 Xe<sub>La</sub>TeX。
2. 先編譯一次樣板或是教學檔中的**根文件 (main.tex)**，確認正常編譯。**編譯前先修改 config.tex 中的中文字型或作業系統，詳情見章節 2.4.2。**<sup>5</sup>如果編譯出錯的話建議自行找人解決，沒人解決可以找我。不過畢竟不在現場，處理上會較為麻煩。自行解決後請回報問題及解法。
3. 依樣畫葫蘆。
  - (a) 設定（各學校目錄中的）config.tex 中的個人資訊以及相關設定。
  - (b) 照教學檔試著插入章節、文字、圖檔、表格.....（圖片檔、程式碼檔案請置入 figures、codes 目錄中。）
  - (c) 須要新增章節子檔的話直接複製 chapter\_template(\_demo).tex，修改內容、重命名後插入 main.tex 對應的章節位置。
  - (d) 要新增使用套件或其他導言區指令請填入 macros\_preamble.tex 檔案，樣板會自動讀取。
  - (e) 同樣的，文件內的共用指令（填在\begin{document}之後的）填入 macros\_document.tex。
  - (f) 若有風格不符合的部份（如：文獻引用要用作者名、列表要照字母排序等等），請於 cls 檔中搜尋關鍵字（編輯器通常會提供，快速鍵 Ctrlf），找到設定行之後自行修改。
4. 全部完成後編譯 main.tex，產生 pdf。
5. 口試後將審定書、推薦書等文書簽名後掃描後存成 pdf 置換原先插入的單頁 pdf。（附件區域已嘗試自動生成客製化文件，省去另開 Word 編輯文件的功夫。）

<sup>5</sup>這部份其實可以透過 ifplatform 套件自動解決，但須要加入“-shell-escape”參數至編譯指令，我怕新手對這比較怕，所以改用設定值處理。

論文各文件、章節次序已經按照學校要求排好，各位只要直接修改或替換即可。

## 1.6 樣板不符合我的要求怎辦？

要新增使用套件或其他導言區指令請填入 `macros_preamble.tex` 檔案，樣板會自動讀取。文件內的共用指令（填在`\begin{document}`之後的）填入 `macros_document.tex`。一些設定要改的話可於 `cls` 檔中搜尋關鍵字（編輯器通常會提供，快速鍵 `Ctrl+f`），找到設定行之後自行修改。

## 1.7 有其他教學嗎？

$\text{\LaTeX}$  教學很多，但大多是英文的。免費資源中，wikibook 的  $\text{\LaTeX}$ <sup>6</sup> 是我非常推薦的，同時也是本文主要參考的文件。中文的完整文件有  $\text{\LaTeX}$  123<sup>7</sup>。不過文中使用的是  $\text{\LaTeX}$  + CJK 套件，而本文用的是  $\text{\XeLaTeX}$ ，少數內容與本樣板不同。

如果你碰到問題，研究半天無解，想問人沒人問又不想上英文論壇問，可以上 ptt 的 latex 板試試。（那裡臥虎藏龍啊！）

## 1.8 MATLAB 圖要如何插入？

不管你是用 MATLAB 還是 Origin 還是其他軟體生成的圖，都要匯出成一般圖檔才能插入。建議格式依序為 `eps`、`png`、`jpg`。<sup>8</sup> 如果是流程圖或一些手繪線條圖，雖然  $\text{\LaTeX}$  有套件可以畫出，但我建議初學者還是用 Dia/Libreoffice Draw 之類的，先畫完再匯出成 `eps` 插入比較快。

## 1.9 編譯出錯怎辦？

看錯誤訊息囉！如果自己找不出解決辦法，試著把錯誤訊息拿去 google 看看。

---

<sup>6</sup><https://en.wikibooks.org/wiki/LaTeX>

<sup>7</sup><http://www.cs.pu.edu.tw/~wckuo/doc/latex123/node1.html>

<sup>8</sup>`eps` 是向量圖，縮放不失真，同時也是  $\text{\LaTeX}$  原本就支援的格式。`png`、`jpg` 則是  $\text{\XeLaTeX}$  才支援。



## 二、 軟體安裝

目前主要針對 Linux、Windows 下的軟體做說明。(Mac 也有相對應軟體，但僅稍微提及)

### 2.1 TeX 編輯器

雖然 TeX 檔均為純文字檔，用任何一個文字編輯器均能編輯（即使是 windows 的記事本）。但選用一個好的 Tex 編輯器可以省下不少功夫，像是快速鍵編譯、語法高亮度、原始碼片段插入，也會比爛爛的 Windows 記事本還安全，像是自動儲存等等。以下列表前兩者為我推薦的編輯器：

1. TexStudio，跨平台的編輯器，具有許多好用功能，並且仍在活躍開發中，非常推薦<sup>1</sup>。(此為 Texmaker 的分支版)
2. Kile，KDE 下的編輯器，我用的是這個。
3. Gummi，一個可以即時編譯的編輯器。
4. TeXShop，Mac 上的另一個選擇。(沒用過，自己試)

關於編譯時的行為，TexStudio 與 Kile 兩者都能夠自動找到根文件（將整份文件分割成不同 tex 檔撰寫時，最上層的主文件）進行編譯。TexStudio 毋須任何設定，Kile 則須要將檔案加入專案中。此功能好處是編輯任何子文件後，直接按下編譯快速鍵即可完整編譯整份文件，不過缺點是浪費時間。如果你有用到 subfile/standalone 等輔助套件可以考慮禁用此功能，也不須修改根文件的內容。(我在 Kile 的作法是除了 main.tex 外，全部加入專案做管理，按編譯後由於 kile 找不到引用的上層文件，所以就會只編譯當前檔案。)

---

<sup>1</sup>Linux/Windows/Mac 均有。也是眾多 Macer 的選擇

## 2.2 TeX 發行版

所謂「發行版」指的是特定組織將編譯 TeX 所須的環境，包含編譯器、常用套件等等包裝在一起發佈的軟體集合。最流行的三個分別是 TeX Live、MiKTeX、MacTeX。TeX Live 設計上是朝多人使用設計（應該是指安裝在伺服器上共用），而 MiKTeX 則是單人使用，所以在安全性方面 TeX Live 會較好，詳情請自行 google。以下是其官方網站：

- TeX Live <https://www.tug.org/texlive/> Linux/Unix, Windows
- MiKTeX <http://miktex.org/> Windows
- MacTeX <http://www.tug.org/mactex/> Mac

LaTeX 網站上對此亦有詳細列表，甚至包含了幾個線上版的 TeX (<https://www.latex-project.org/get/>)。下面介紹不同作業系統下的安裝方法。

### 2.2.1 GNU/Linux

一般來說 Linux 儘量由 (Linux) 發行版自己的套件庫安裝會比較好，這裡以 Debian 的指令模式舉例，視窗介面或是其他發行版請自行舉一反三。

以下是我在 Debian 中須要手動安裝的套件，其他須要安裝的會透過相依性自己安裝。(如果想全裝就裝 texlive-full 套件)

套件名稱 (in debian)	說明及需求 TeX 套件
texlive	主要套件，會連帶安裝基本系統
texlive-bibtex-extra	BibLaTeX
texlive-lang-chinese	xeCJK，連帶安裝 CJK 相關套件
texlive-science	siunitx (單位)
texlive-xetex	X <sub>Y</sub> LaTeX
biber	搭配 BibLaTeX

對應的一行安裝指令為

```
apt-get install texlive texlive-bibtex-extra texlive-lang-chinese
```

稍微注意一下 lmodern 套件是否自動安裝，我在 debian 7 上面碰過沒自動裝，導致找不到字型的問題。(這套件包含一些英文 PS 字型，可能是

TeX Live 2012 預設字體使用到 lmodern)

稍微說明一下，我的樣板使用  $\text{X}_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  + Biblatex + biber。如果要  
用傳統的 Bibtex 做文獻索引的話免裝 texlive-bibtex-extra biber。

缺套件時除了 google 後安裝含所須 tex 套件的 Linux 套件這個方法外，  
也可以使用 Tex Live 自己的套件管理程式 tlmgr 來安裝單一套件。

### 2.2.2 Windows

請上各 L<sup>A</sup>T<sub>E</sub>X 發行版官方網站下載。安裝內容若不知如何選請按完整  
安裝或是日後透過 Tex Live 自己的套件管理程式 tlmgr 進行安裝（目前的  
TeX Live 全裝約 4.5GB）。MiKTeX 具有即時安裝的功能，編譯時若有缺  
會自動下載。

## 2.3 文獻資料庫 (.bib) 管理軟體

如果你的文獻列表使用到文獻資料庫 (.bib) 的話，建議你安裝一個  
管理軟體比較方便。這節的說明若看不懂可以先跳到基礎教學的文獻 (章  
節 3.11) 看一下。先列出常用的文獻管理軟體：

**Zotero** 極度推薦！同時也是我使用的文獻管理軟體。免費、自由 (AGPL  
授權)、開源、跨平台、Unicode(可引入中文條目)。加上 BBT 擴充後  
可以方便的插入 `\cite{cite_key}`。(下面專門介紹)

**JabRef** 使用 Java 開發，免費、自由 (MIT 授權)、開源、跨平台、似乎支  
援 Unicode，但須手動設定。可以方便的插入 `\cite{cite_key}`。

**Bibus** 很早期的 Linux 軟體介紹會出現，但個人感覺頗難用，且已停止開  
發。

**KBibTeX** KDE (Linux 的桌面環境之一) 提供的漂亮軟體。沒用過 XD，  
用 KDE 有興趣玩玩。

**EndNote** \$\$\$。非常昂貴的軟體（我知道各位的學校、研究單位都已經  
很好心的幫我們繳錢了，這點可能不是問題...）。不支援中文、不支援  
Linux。個人不推。

我自己用的是 Zotero，所以在這裡詳述心得。Zotero 有 Firefox addon 版以及 Standalone 版<sup>2</sup>。我是兩個都裝，因為 Firefox addon 版可以在瀏覽網頁的當下點 addon 提供的按鈕，直接匯入網頁、期刊的文獻資料，省去慢慢手打的時間。這方面其他文獻管理軟體就沒這麼方便。<sup>3</sup>。而 Standalone 版的好處則是操作快速，當然這可能跟我的 Firefox 有 1000 個分頁有關，各位可自行決定是否加安裝 Standalone 版。

L<sup>A</sup>T<sub>E</sub>X 引用文獻時須要使用 `\cite{cite_key}` 這樣的指令，但對於大量文獻，我們很難記得每一個的 `cite_key`（有些編輯器支援自動補完，但仍須記得前幾個字母），所以文獻管理軟體要是能支援插入 `\cite{cite_key}` 是一件非常美好的事，這也是我朋友極力推薦 JabRef 的原因之一。Zotero 雖然本身不提供這個功能，但透過 **Better Bib(La)TeX** 擴充後能夠以拖曳方式插入 `\cite{cite_key}`。

安裝完 Zotero 以及 BBT 套件<sup>4</sup>後，開啓 Zotero 設定視窗，在「匯出」分頁的「預設輸出格式」選「Better BibTeX Quick Copy」即可使用拖曳插入 `\cite{cite_key}` 的功能。

最後要編譯前記得對 Zotero 中的文獻庫（左方窗格的資料夾圖示）按滑鼠右鍵，選「匯出文獻庫」並選擇「BibLaTeX」格式（如果你打算採用 BibTeX 格式請記得自行修改樣版）匯出 .bib 文獻庫。<sup>5</sup>

---

<sup>2</sup>亦有 Chrome 和 Safari 版，見<https://www.zotero.org/download/>

<sup>3</sup>期刊可以使用 google scholar 各條目下的「引用」複製所須資訊（如 BibTeX、EndNote）再匯入管理軟體，。EndNote 本身則是提供搜索引擎，可以直接取得相關資訊。

<sup>4</sup>BBT 擴充的專案位於<https://github.com/retorque/zotero-better-bibtex>

<sup>5</sup>JabRef 這類直接對 .bib 進行管理的軟體應該就不須要再匯出了。

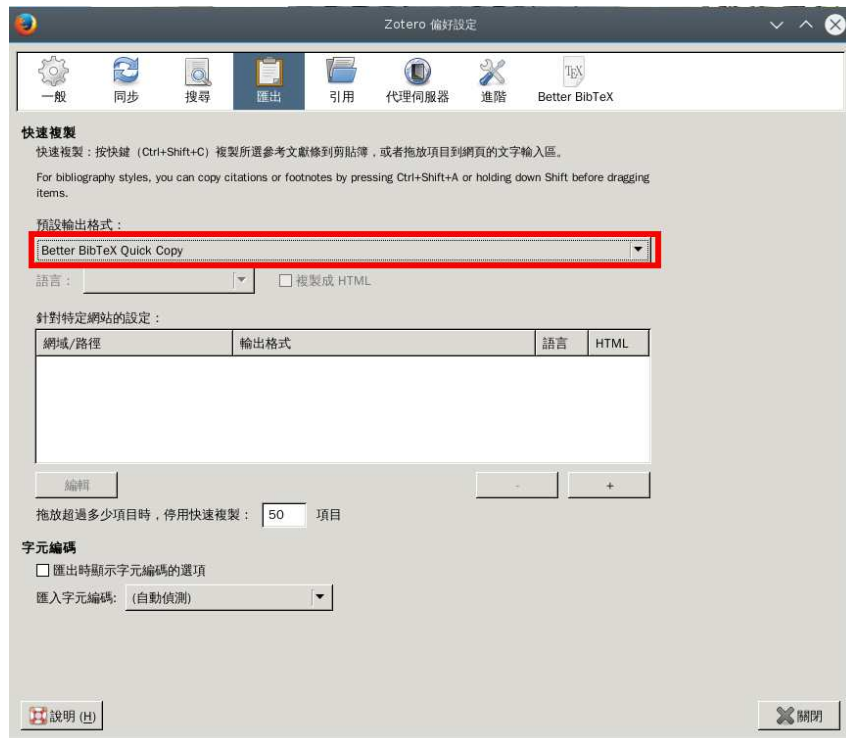


圖 2.1: BBT 拖曳插入 cite\_key 的設定方法。

## 2.4 本 L<sup>A</sup>T<sub>E</sub>X 專案結構及使用說明

### 2.4.1 取得專案

本專案公開於 GitHub，你可以在“TW\_Thesis\_Template”專案首頁（不是我的帳號首頁喔！）右邊找到 Download 鈕（圖 2.2）下載整個專案（github 不提供下載單一目錄。單檔可以進入頁面後另存）。下載前確認你要載的版本，左側“Branch”鈕可以選擇不同的分支（圖 2.3a），master 分支通常是穩定版，而有時為了測試會新增 develop 分支。“tag”則是在重要版本發布時（Ver 1.0）才會特別標記（圖 2.3b）。



圖 2.2

## 二、軟體安裝



圖 2.3: GitHub 的分支及標籤列表。

### 2.4.2 專案結構

專案根目錄包含教學（LaTeX tutorial）、各校規定（info）、各校樣板與語系（NCU\_zh）三類型目錄。另附上教學與樣板的編譯結果 (\*.pdf) 作為閱讀以及對照使用者編譯之用。各 LaTeX 目錄中的共用檔案及目錄 (結尾含 “/” 者) 如下，請盡量不要修改檔名或刪除檔案：，config.tex 用來設定個人資訊

codes/	用來存程式碼。
figures/	用來存放論文中的圖片。亦可直接存在根目錄，但可能顯得凌亂。
NCU_thesis.cls	樣板主要設定，基於 LaTeX book class，須要修改格式可以由此檔修改。
main.tex	主檔案。 <b>欲生成整份文件請編譯此檔。</b> 檔案用於定義各子檔（章節）順序。
config.tex	個人資訊設定檔。設定如論文題目等設定。 <b>請設定使用之中文字型再編譯。</b> 修改的方法是直接將花括號 {} 內改成你要的設定值。 <sup>a</sup> ，不然可能因找不中文字型而出錯。 <b>本樣板可以利用設定作業系統來使用各系統預設字型。</b>
macros_preamble.tex	共用導言檔。將你要放在各個檔案導言區 <sup>b</sup> 的指令置於此檔。此檔將由 NCU_thesis.cls 讀入，子文件中不須再引入。
macros_document.tex	共用文內設定。有些指令必須置於文內（\begin{document} 內），如設定字體大小的 fontsize 指令。這類型指令可置於此檔保持各子檔整潔以及避免漏放指令。
titlepage.tex	封面。
acknowledgements.tex	誌謝（可不加，自行於 main.tex 註解掉）。
abbreviation.tex	符號表。這裡簡單的以表格列出。 <sup>c</sup>
abstract_*.tex	各語系摘要。
chapter_*.tex	各章節。（通常一章就一個檔。當然你可以再細分成更小的 section 子檔。 <sup>d</sup> ）
appendix_*.tex	各附錄。
letter_*.pdf	各校須附文件。（不附加直接刪除即可，可不註解）

上面共用檔名開頭（如 chapter\_\*.tex）只是我個人為了文件目錄的整潔而自訂的規則，各位自行決定是否跟隨。

## 三、 $\text{\LaTeX}$ 基礎教學

這裡的教學目的只是希望  $\text{\LaTeX}$  新手可以無痛寫論文。所以只說明搭配此樣版後須要具備的基礎知識。如何調整樣板不是主要目標，有興趣或是有此需求的可以直接看  $\text{\LaTeX}$  code，內有一些註解或網頁連結可能可以幫到你。下面的說明我儘量以範例呈現，各位可以直接複製所須的  $\text{\LaTeX}$  片段貼上論文再改。不過由 pdf 複製文字的部份怪怪的，可能要由  $\text{\LaTeX}$  原始碼複製。這部份可能是 pdf cmap 等等的問題。

### 3.1 背景知識

#### 3.1.1 $\text{\TeX}$ 、 $\text{\LaTeX}$ 歷史

$\text{\LaTeX}$  是一個非常強大、專業的排版軟體。事實上  $\text{\LaTeX}$  其實是  $\text{\TeX}$  的巨集。它使得原本較為困難的  $\text{\TeX}$  語法變得簡單許多。 $\text{\TeX}$  是 Donald Knuth 1977 年為其巨著「The Art of Computer Programming」<sup>1</sup> 進行校稿時，發現當時的數學公式排版過於差勁，於是決定自己弄一個排版系統出來（這才是真正的 Hacker !!）。由於直接寫  $\text{\TeX}$  或稱 Plain  $\text{\TeX}$  文件較困難，因此後來發展出  $\text{\LaTeX}$  來簡化上手難度。而後來又因為非英語、直接產生 PDF 等需求又衍生出各式各樣的  $\text{\TeX}$  / $\text{\LaTeX}$  版本。<sup>2</sup>

#### 3.1.2 編譯/生成 PDF 檔

$\text{\LaTeX}$  不同於 LibreOffice Writer/ MS Office Word 之類的程式，它須要寫好 tex 檔之後以 Latex 程式（再透過  $\text{\TeX}$  程式）編譯成最終檔案<sup>3</sup>。其中 tex 檔為純文字檔，和常見的.txt 文字檔相同，亦即你可以由任何文字

<sup>1</sup>這本「神書」目前還沒寫完全套，希望 Knuth 有生之年可以順利完成。

<sup>2</sup> $\text{\TeX}$  追求穩定，所以只除錯，不添加新功能。

<sup>3</sup>原始的  $\text{\LaTeX}$  編譯器（就稱為 latex）是將 tex 轉為 dvi 檔（可用 evince 等軟體開啓）。若要再轉 pdf 須要再經由其他軟體轉換。

編輯器開啓，如 Linux 上的 VIM 或是 Windows 上的記事本。相對的像是 Word 的 doc 檔一定要用支援 doc 的程式才能開啓，否則會顯示成亂碼。

$\text{\LaTeX}$  編譯器除了本身稱為  $\text{\LaTeX}$  的程式外，尚有  $\text{\XeTeX}$ 、 $\text{\pdfLaTeX}$ 、 $\text{\LuaLaTeX}$  等編譯器<sup>4</sup>。其特色如下：

**$\text{\XeLaTeX}$**  支援 Unicode 萬國字碼 (原生 UTF-8)。支援現代系統中的常見字型格式，如 TrueType/OpenType，也就是說不須要另外載  $\text{\LaTeX}$  用的 PostScript 字型。輸出檔為 PDF (但不支援 DVI/PS)。

**$\text{\pdfLaTeX}$**  非 Unicode，輸出檔為 PDF。中文字碼處理須透過 CJK 套件，似乎以奇怪方式支援<sup>5</sup>。(其他優點我就不知啦！)

**$\text{\LuaLaTeX}$**  人如其名； $\text{\LuaLaTeX}$  即是基於  $\text{\pdfLaTeX}$  再加上 Lua 腳本引擎。由於  $\text{\TeX}$  並不是真正的程式語言，要實現某些程式化動作可能須要拐彎抹角甚至是無法達成。而加上 Lua 則能夠以真正的程式語言優美的完成所須動作。支援 Unicode 萬國字碼。

目前看起來  $\text{\XeTeX}$  是較為先進與穩定的編譯器，因此本文僅針對  $\text{\XeTeX}$  說明以及測試。(教學部份一般適用於所有  $\text{\LaTeX}$  編譯器，若你使用其他編譯器應該也只有少數區域要修改。)

$\text{\LaTeX}$  編譯時不會一次完成所有的工作，因此須要多次編譯。下面是含 .bib 文獻的標準編譯順序<sup>6</sup>。(main 表示 main.tex，為某一專案的主檔案)

1. `xelatex main`  $\leftarrow$  先編譯一次，部份內容如文獻以代號紀錄下來。
2. `biber main`  $\leftarrow$  處理 .bib 文獻資料庫。`biber` 是  $\text{\BibLaTeX}$  用的程式，若你用  $\text{\BibTeX}$  則為 `bibtex main`
3. `xelatex main`  $\leftarrow$  將上次編譯產生的代號替換為最終格式。
4. (`xelatex main`)  $\leftarrow$  可能視情況還須要更多次編譯 (如 `makeindex`)。

---

<sup>4</sup>他們相對應的  $\text{\TeX}$  編譯器分別是  $\text{\XeTeX}$ 、 $\text{\pdfTeX}$ 、 $\text{\LuaTeX}$ 。事實上它們的特色是由  $\text{\*TeX}$  所提供而非  $\text{\*LaTeX}$ 。

<sup>5</sup><https://www.csie.ntu.edu.tw/~tzhuan/www/resources/ntu/>

<sup>6</sup>若須生成目錄 (樣板 main.tex 會生成目錄)，`\tableofcontents` 指令，須編譯到第二次才會生成。



通常 L<sup>A</sup>T<sub>E</sub>X 專用編輯器已經幫你處理好了，你只須要按下快速鍵（或圖示）一次就可以編譯完成。

指令介面的使用者對於單次編譯可以使用下面指令

```
xelatex -interaction=nonstopmode main
```

而多次編譯的話可以安裝 latexmk 這個軟體，它可以自動完成所須的編譯動作。<sup>7</sup> X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X 使用者可以使用以下指令：

```
latexmk -xelatex -interaction=nonstopmode main
```

編譯完成後，若你的系統上有 Ghostscript (Linux 上應該很常見) 可以以下面指令壓縮 pdf 檔：

```
gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -sOutputFile="Compressed.pdf" "Original.pdf"
```

你可能會發現在編譯之後，工作目錄下出現許多奇怪附檔名的檔案。這是由於 L<sup>A</sup>T<sub>E</sub>X 編譯時無法往回處理之前的標記，所以須要將標記資料紀錄下，待下次編譯時再處理。你可以不理會它們，對你而只有.tex（文件檔）、.bib（你置入的文獻資料庫）以及編譯生成的.pdf 須要保留。不過有時會因為殘留的紀錄檔導致編譯出錯，所以有時遇到奇怪問題時記得試試先清除它們再編譯，或是直接將「清除紀錄檔」這個動作加到編譯序列的最前面。<sup>8</sup>

## 3.2 文件結構

最簡單的 L<sup>A</sup>T<sub>E</sub>X 文件如下：

```
1 \documentclass{article}
2 % This is comment, here is preamble.
3 \begin{document}
4 Hello world!
5 \end{document}
```

以\開頭的均為指令，而{}為其選項。\\documentclass 選擇文件的類型(class)，article 適合一篇簡單的文件(如實驗報告)。而其他如 book/report 則適合較大型的書、期刊等等。本樣板即是使用 book class。\\begin ~\\end

<sup>7</sup>latexmk 似乎會自行處理標記(\\label)，所以我章節 3.5提到的快速插圖指令\\insfig 會顯示找不到標記。加上“-f”強制繼續編譯即可，但要注意是否有其他「真正」的錯誤導致編譯出錯。

<sup>8</sup>各檔案說明見[https://en.wikibooks.org/wiki/LaTeX/Basics#Ancillary\\_files](https://en.wikibooks.org/wiki/LaTeX/Basics#Ancillary_files)

則是某個環境的範圍。(這裡是 document 環境) `\begin{document}` 表示其中為文件內容。而 `\begin{document}` 前面的區域為導言區 (preamble)，用來設定所用的套件。

各個不同的文件的類型 (class) 提供不同的風格設定。以本樣板用到的 book 為例，其原始結構如下：

```

1 \begin{document}
2 \frontmatter
3
4 \maketitle % print titlepage or cover
5 \tableofcontents % table of contents (目錄)
6 \listoffigures % list of figures (圖片列表)
7 \listoftables
8
9 % Introductory chapters
10 \chapter{Preface}
11 % ...
12
13 \mainmatter
14 \chapter{First chapter}
15 % ...
16
17 \appendix
18 \chapter{First Appendix}
19
20 \backmatter
21 \chapter{Last note}
22 \printbibliography % print bibliography in here. (文獻)
23 \end{document}

```

各區段 (frontmatter/mainmatter/backmatter/...) 有不同設定，如頁碼的差異、章節的編號等等。`\chapter` 表示新的一章。類似的像是 `\section`、`\subsection` 表示節、小節。<sup>9</sup>

依照中央大學的要求，`\appendix` 必須在 `\backmatter` 後面<sup>10</sup>，因此 NCU 樣板有對 `\appendix` 做一點小小修改，使其置於 `\backmatter` 之後亦正常顯示。

### 3.3 套件與指令

$\text{\LaTeX}$  文件中，提供各種特別功能的就稱為指令，而指令有些須要另載入「套件」才会有。譬如要打出水的化學式  $\text{H}_2\text{O}$ ，你須要載入 `mhchem` 套件<sup>11</sup>，然後利用它提供的指令：`\ce{H_2O}`。

<sup>9</sup>樣板預設目錄顯示至第三級 (`\subsection{}`)，章節編號標至第五級 (`\paragraph{}`)。

<sup>10</sup>據說 APA 格式亦要求 `\appendix` 放至最後。

<sup>11</sup>我知道有些人是直接數學模式啦！

要載入套件可於導言區 (preamble) 使用：

```
1 \usepackage[option]{package}
```

而各指令的形式如下

```
1 \commandname[option1,option2, ...]{argument1}{argument2} ...
```

指令起始於`\`。有些指令不帶參數，如`\LaTeX`<sup>12</sup>。有些指令則是須要一個以上的參數，其中`{argument}`為必要參數。`[option]`為可略選項。譬如圖片說明文字的指令為：

```
1 \caption[short]{full}
```

當填入`[short]`時，圖列表處的說明會換成短版的說明（顯示於圖目錄）。

`\`起始的指令名稱直到空格或保留字才結束，因此對於內文中的無參數指令一定要加空格或是以`{}`結束。下面示範於內文中（以`text`夾住）使用指令

表 3.1: 指令範例

<code>text\cmd_text</code>	o Valid
<code>text\cmd{}text</code>	o Valid
<code>text\cmdtext</code>	x Invalid
<code>text\cmd文字</code>	x Invalid

有些指令提供帶星號形式。如`\chapter`與`\chapter*`，後者將產生一個無編號的章節，同時也不顯示於目錄。

指令的參數（`[]`或是`{}`）內不能有多餘的空格。譬如插入圖片時的路徑若在結尾多打一個空格：

```
1 \includegraphics{logo-Linux.png }
```

編譯時會出現錯誤，因為編譯器會將空格視為檔名，所以找不到檔案。

## 3.4 文字

$\text{\LaTeX}$  中的文字編排有以下特性：

- 除了保留字外，一般文字直接打就可以顯示在文件上。

<sup>12</sup>`\LaTeX` 用在文字行時，結尾要加上反斜線、空格：`"\LaTeX\ "` 不然會和後面黏在一起。

- 區分大小寫。`\chapter` 與 `\Chapter` 是不一樣的。
- 縮減空格。文字間多個空格與單一空格等效，僅會顯示出同樣的間隔，而此一間隔由 L<sup>A</sup>T<sub>E</sub>X 編譯器判斷。
- 忽略單一斷行。單一斷行與連接在一起是等效的。二個以上的斷行視為新的段落。若要強制斷行但不分段可以使用雙反斜線：`\\`。
- 英文“引號”，左邊要用兩個「```」（曲線號「`~`」那顆鍵，1 左邊），右邊要用兩個「`'`」（Enter 左邊的鍵）

### 3.4.1 保留字

以下字元為保留字，直接使用不會出現在文件上 (大部分會導致編譯失敗)。

保留字	顯示文字指令	保留字功能
<code>#</code>	<code>\#</code>	定義指令的選項
<code>\$</code>	<code>\\$</code>	數學模式標記
<code>%</code>	<code>\%</code>	註解
<code>^</code>	<code>\^{}</code>	上標，用於數學模式
<code>&amp;</code>	<code>\&amp;</code>	矩陣、表格分隔符
<code>_</code>	<code>\_</code>	下標，用於數學模式
<code>{</code>	<code>\{</code>	指令參數標記
<code>}</code>	<code>\}</code>	指令參數標記
<code>~</code>	<code>\~{}</code>	
<code>\</code>	<code>\textbackslash{}</code>	指令起始字元

### 3.4.2 字體大小

字體大小預設 12pt(1pt=1/72 inch)，設定在檔案最開頭如：

```
1 documentclass[a4paper,12pt]{book}
```

不過 12pt 對於中文來說太小，而 L<sup>A</sup>T<sub>E</sub>X 預設文件類別又只提供有限的選項，且最高僅到 12pt。解決方法有兩種。一是在內文區段 (`\begin{document}` 內) 使用 `\fontsize` 調整。

```
1 \fontsize{size}{baselineskip}\selectfont
```

L<sup>A</sup>T<sub>E</sub>X 預設值 10pt,12pt。本樣板設定 14pt,25pt (約 1.5 倍行高)。

另一個則是以其他文件類別如 `scrbook` 來取代傳統的 `book class`。`scrbook` 提供了許多方便的選項以及特性，有興趣可以試試看。由於本樣板原先就使用 `book` 為基底，且考慮到有些舊的 TeX 發行版可能未包含 `scrbook`，因此目前暫時不轉移至 `scrbook`。

對於區域性的修改字體大小指令示範如下<sup>13</sup>：

表 3.2: 字體大小範例

指令	效果
<code>\tiny{...}</code>	字體大小
<code>\normalsize{...}</code>	字體大小
<code>\large{...}</code>	字體大小
<code>\LARGE{...}</code>	字體大小
<code>\Huge{...}</code>	字體大小

### 3.4.3 字型樣式

L<sup>A</sup>T<sub>E</sub>X 中的字形樣式以指令設定，包含文字風格（粗、斜體）以及字型。下面這張表格示範了常用的樣式指令以及效果。其中第一欄 (L<sup>A</sup>T<sub>E</sub>X) 指令會影響後面的所有文字<sup>14</sup>。第二欄 (小範圍) 則是只影響 `{...}` 中的內容。最後面分別測試中英文的效果。<sup>15</sup> <sup>16</sup> 本樣板中，`\rmfamily`, `\sffamily`,

表 3.3: 樣式範例

L <sup>A</sup> T <sub>E</sub> X	小範圍	效果
<code>\normalfont</code>	<code>\textnormal{...}</code>	正常 normal
<code>\em</code>	<code>\emph{...}</code>	強調 <i>emphasis</i>
<code>\rmfamily</code>	<code>\textrm{...}</code>	羅馬 (襯線) roman
<code>\sffamily</code>	<code>\textsf{...}</code>	無襯線 sans serif
<code>\ttfamily</code>	<code>\texttt{...}</code>	打字機 (等寬) teletypefont (monospace)
<code>\bfseries</code>	<code>\textbf{...}</code>	粗體 bold
<code>\itshape</code>	<code>\textit{...}</code>	斜體 <i>italic</i>

`\ttfamily` 可以由 `config.tex` 的 `\def\mainfont` 等指令設定使用的字型，中文則是 `\def\CJKmainfont` 等指令。<sup>17</sup>

<sup>13</sup>完整列表見 [wikibook:https://en.wikibooks.org/wiki/LaTeX/Fonts#Sizing\\_text](https://en.wikibooks.org/wiki/LaTeX/Fonts#Sizing_text)

<sup>14</sup>可使用 `{}` 來限制範圍，如 `{\em ...}`

<sup>15</sup>改自 L<sup>A</sup>T<sub>E</sub>X [wikibook:https://en.wikibooks.org/wiki/LaTeX/Fonts#Shapes](https://en.wikibooks.org/wiki/LaTeX/Fonts#Shapes)

<sup>16</sup>底線須要另外使用 `ulem` 套件，理由是應以 `emphasis` 取代，見 [wikibook](#)。

<sup>17</sup>事實上是透過 `fontspec` 套件提供的 `\setmainfont`, `\setsansfont`, `\setmonofont` 進行設定。而對應的中文則是由 `xeCJK` 套件提供的 `\setCJKmainfont`, `\setCJKsansfont`, `\setCJKmonofont`

普通文件常用的字體有楷體以及明體 (或用宋體) 兩種。通常內文較適合細字的明體，這樣讀起來較不容易疲勞。而楷體適合放在字體、空間較大的標題，因此本樣板中設定主要字體為明體，而無襯線字體則是設定楷體。若要新增其他字型，可以使用`\newfontfamily` 來設定。下面是一個簡單的範例，這裡假設你的系統中有「`cwTeX` 楷書」字型。

```
1 \begin{document}
2 \newfontfamily\cwbm{cwTeX 楷書}
3 正常文字
4 {\cwbm 使用 cwTeX 楷書}
5 正常文字
6 \end{document}
```

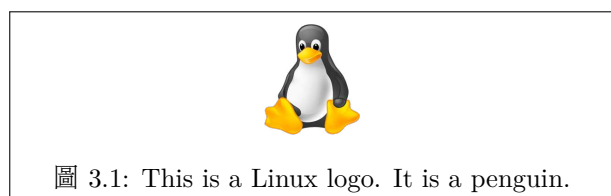
有些文件會以一種較短的指令來設定樣式，如粗體使用`\bf`、斜體使用`\it`。這是舊的指令。現在 (2016 年) 請用`\bfseries`、`\itshape`。

## 3.5 圖片

$\text{\LaTeX}$  支援的圖片格式有 eps, pdf, png, jpg。數據圖推薦使用 eps 格式，因為 eps 是向量圖檔，縮放不會失真。流程圖或一些手繪線條圖，雖然  $\text{\LaTeX}$  有套件可以畫出，但我建議初學者還是用 Dia/Libreoffice Draw 之類的<sup>18</sup>，先畫完再匯出成 eps 插入比較快。<sup>19</sup>

這裡用的圖片全部都不採用「文繞圖」模式，一般正式文件應該也很少使用文繞圖。欲使用的話請自行 google。

### 3.5.1 插入單一圖片



```
1 \begin{figure}[!htb]
2   \centering
3   \includegraphics[width=0.2\textwidth]{
4     logo-Linux.png}
5   \caption[Linux logo]{This is a Linux
6     logo. It is a penguin.}
7   \label{fig:first_linux_logo}
8 \end{figure}
```

注意幾件事：

- `{figure}` 環境為一個浮動性環境，也就是說在沒有使用 `[h]` 選項的情況下， $\text{\LaTeX}$  會自動為我們選擇適當位置。據我個人觀察，它會盡量不切

<sup>18</sup>推薦的自由軟體：影像、照片修圖：GIMP。向量圖、手繪線條：Inkscape、Libreoffice Draw。流程圖：Dia

<sup>19</sup>M\$ Office「力點」應該也可以啦！我沒試過。

斷文字，並放置於頁的上、下方，再來是直接放置於同一頁。雖然說一般文章會建議不要使用 `h(here)` 選項強制置於指令位置，但有時  $\text{\LaTeX}$  自動排列會出現一些非常愚蠢的結果，可能還是須要自行決定是否使用 `h`。<sup>20</sup>

- `\includegraphics` 中，`width=0.2\textwidth` 指得是縮放圖形至 0.2 倍行寬。
- `\label` 須要放在 `\caption` 之後，因為 `\caption` 存在 `\label` 才會生成標記。

若只是插入一張圖片，上面的指令稍嫌麻煩，於是我提供一個簡化指令，其語法如下：

```
1 \insfig[scalefactor][label][!htb]{path}[caption][short caption]
```

除 `path` 外，其餘均可忽略，但 `[]` 中間選項忽略時要補上空括號，以下照優先順序排列說明。

1. `path`，圖片路徑含副檔名。
2. `scalefactor` 縮放因子， $0.0 \sim 1.0$  倍行寬，預設 0.5
3. `label` 標籤，省略則預設「fig:」+ 檔名 (`path`)
4. `!htb` 浮動選項，預設為空，即交由  $\text{\LaTeX}$  排版。
5. `caption`，圖片說明，顯示在 `\includegraphics` 的相對位置。
6. `short caption` 短圖說，用來取代圖目錄裡的文字，預設為空，即同 `caption`。

須要注意的是括號僅 `path` 為大/花括號，這是因為路徑為必要參數。以下示範各種使用方法：

<sup>20</sup>本樣板已於 `cls` 中設定 `FloatBarrier`，浮動環境不會超出 `\subsubsection` 範圍。



圖 3.2



圖 3.3: line 2, caption



圖 3.4



圖 3.5: line 4, caption



圖 3.6: line 5, caption



圖 3.7: line 6, caption



圖 3.8: line 7, caption

```

1 \insfig{logo-Linux.png}
2 \insfig{logo-Linux.png}[line 2, caption]
3 \insfig[0.15]{logo-Linux.png}
4 \insfig[0.15] [] [h]{logo-Linux.png}[line 4,
  caption]
5 \insfig[0.15] [fig:labal_test_1] [h]{logo-
  Linux.png}[line 5, caption]
6 \insfig[0.15] [fig:labal_test_2]{logo-Linux.
  png}[line 6, caption]
7 \insfig[0.15] [fig:labal_test_3] [!hbt]{logo-
  Linux.png}[line 7, caption] [short
  caption]

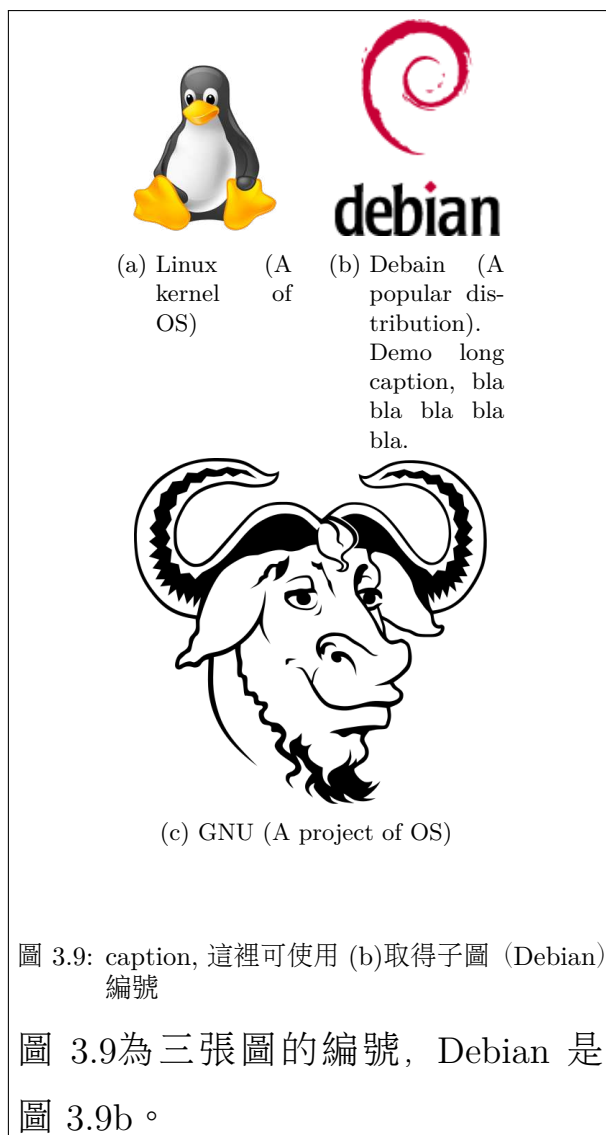
```

省略 labal 的圖會自動設定 labal 為 “fig:logo-Linux.png”

### 3.5.2 插入多張圖片

這裡使用 subcaption 套件並且使用 `\subcaptionbox`，它可以對齊各子圖的底部，多行 caption 時比較整齊。下面範例同時示範文內引用的方法。





```

1 \begin{figure}[!hbt]
2   %\captionsetup[subfigure]{labelformat=
3   empty} % 完全隱藏圖號
4   \centering
5   \subcaptionbox
6     {Linux (A kernel of OS)}
7     {\label{fig:subfig_linux}}
8     {\includegraphics[width=0.3\linewidth]{logo-Linux.png}}
9   ~
10  \subcaptionbox
11    {Debain (A popular distribution).
12    Demo long caption, bla bla bla bla.}
13    {\label{fig:subfig_debian}}
14    {\includegraphics[width=0.3\linewidth]{logo-Debian.png}}
15  \vspace{\baselineskip} % 分隔上下列
16  \subcaptionbox
17    {GNU (A project of OS)}
18    {\label{fig:subfig_gnu}}
19    {\includegraphics[width=0.6\linewidth]{logo-GNU.png}}
20    % use ``\subref{fig:subfig_debian}''
21    % get ID of subfigure(this ID is Debian)
22    \caption{caption, 這裡可使用 \subref{fig:subfig_debian}取得子圖 (Debian) 編號 }
23    \label{fig:multifig_labal}
24 \end{figure}
25 % 以下內文示範引用
26 \cref{fig:multifig_labal}為三張圖的編號,
27 Debian是\cref{fig:subfig_debian}。

```

~ 表示在圖片間插入間隔，當圖片較密時比較美觀。使用這個多圖的格式時，圖片一行的寬度總和建議不要超過「 $0.98\text{\linewidth}$ 」，不然會自動折行跑到下一行去。也由於這個自動排列的特性，`\subcaptionbox` 無法做很精確的圖片排列，譬如說第一行 0.3，第二行  $0.3+0.3+0.3$ ，`\subcaptionbox` 一定會將第二、三張圖移到第一行。如果要做這種設定請自行參考 `minipage` 用法。<sup>21</sup>

引用圖片時直接使用 `\labal` 的標記即可，子圖會自動加上編號，如：

<sup>21</sup><http://tex.stackexchange.com/questions/55337/how-to-use-figure-inside-a-minipage>

## 3.6 表格

表格如果太複雜不會製作或是嫌不好輸入，可以找圖形界面的軟體或是線上網站。同樣的，有些編輯器也已經內建表格產生器了，不過功能可能很陽春就是了。表格的說明文字 (caption) 位置改設定在表格上方。(雖然說 caption 位置可以透過 `\caption` 和 `tabular` 環境的相對位置來設定，但產生出的 caption 與表格距離會過近<sup>22</sup>，所以本樣板透過 `floatrow` 套件設定表格的 caption 位置， $\text{\LaTeX}$  code 中 `\caption` 也不必特別寫在 `tabular` 上方。)

表格的 `{table}` 環境和 `{figure}` 環境一樣屬於浮動性環境，詳情見章節 3.5 中的說明。

### 3.6.1 一般表格

表 3.4: Solution	
Component	Concentration(mM)
NaCl	118.0
CaCl <sub>2</sub>	2.11
KCl	4.69

```

1 \begin{table}[h]
2   \centering
3   \caption{Solution}
4   \begin{tabular}{| l | l |}
5     \hline
6     Component & Concentration(mM) \\ \hline
7     \ce{NaCl} & 118.0 \\ \hline
8     \ce{CaCl_2} & 2.11 \\ \hline
9     \ce{KCl} & 4.69 \\ \hline
10    \end{tabular}
11 \end{table}

```

`\begin{tabular}{| l | l |}` 旁的「|」表示畫垂直線，英文字母「l」表示左對齊，`\hline` 表示水平線。對齊控制除了「l」外，亦可用「c」(center)、「r」(right)。表格分隔若想用其他字元亦可用「@{ }」處理，樣板的符號列表就是利用冒號分隔「@{ : }」

### 3.6.2 自動折行表格

利用 `tabularx` 套件使表格自動折行。下面以原始 `tabular` 進行對照。當文字寬度超出表格時，原始 `tabular` 會超出頁面 (下面超出的表格是正常的喔!)，而 `tabularx` 會自動折行以及調整表格。

<sup>22</sup>事實上這個距離由 `\abovecaptionskip` 和 `\belowcaptionskip` 設定，預設為 10pt、0pt。

short	ssssssss
long	lllllllll llllll llllll llllllllll llllll
short	ssssssss
long	lllllllll llllll llllll llllllllll ll- lllll

```

1 \begin{table}[h]
2   \centering
3   \begin{tabular}{|l|l|}
4     \hline
5     short & ssssssss \\ \hline
6     long  & llllllllll llllll llllllllll llllll \\
7     \hline
8   \end{tabular}
9 \end{table}
10
11 \begin{table}[h]
12   \centering
13   \begin{tabularx}{\textwidth}{|l|X|}
14     \hline
15     short & ssssssss \\ \hline
16     long  & llllllllll llllll llllllllll llllll \\
17     \hline
18   \end{tabularx}
19 \end{table}

```

### 3.6.3 複雜表格

示範合併欄位。

1	2	3	d
4	5	66	e
7	8		f
9	a		g
b	c	h	i

```

1 \begin{table}[h]
2   \centering
3   \begin{tabular}{|l|l|l|l|}
4     \hline
5     1 & 2 & 3 & d \\ \hline
6     4 & 5 & \multirow{2}{*}{66} & e \\
7     \cline{1-2}\cline{4-4}
8     7 & 8 & & f \\ \hline
9     9 & \multicolumn{2}{c|}{a} & g \\
10    \hline
11    b & c & h & i \\ \hline
12  \end{tabular}
13 \end{table}

```

### 3.6.4 其他表格

對於太長、太寬的表格可以參考 LaTeX 123 的說明 (<http://www.cs.pu.edu.tw/~wckuo/doc/latex123/node9.html>)

## 3.7 列表/條列

常用的列表形式有項目 (itemize)、數字 (enumerate)、描述 (description) 三種，以下示範項目 (itemize) 列表：

- item 1
- item 2
  - item 2.1     $\leftarrow$  nesting
  - item 2.2
- item 3

```
1 \begin{itemize}
2   \item item 1
3   \item item 2
4   \begin{itemize}
5     \item item 2.1  $\quad \leftarrow$ 
6     nesting
7     \item item 2.2
8   \end{itemize}
9   \item item 3
10 \end{itemize}
```

列表允許巢狀使用，也就是在列表環境中再包一個列表。下面的示範就只列出兩個項目。

數字 (enumerate) 列表：

1. item 1
2. item 2

```
1 \begin{enumerate}
2   \item item 1
3   \item item 2
4 \end{enumerate}
```

描述 (description) 列表：

- description 1** item 1
- description 2** item 2

```
1 \begin{description}
2   \item[description 1] item 1
3   \item[description 2] item 2
4 \end{description}
```

## 3.8 數學符號、方程

數學式的編輯實在太廣，這裡只講解一些基本概念。更多資訊請參考 wikibook: <https://en.wikibooks.org/wiki/LaTeX/Mathematics>

### 3.8.1 顯示模式

$\text{\LaTeX}$  中的常用數學顯示模式有下面三種：

- Equation 環境，由 `\begin{equation}...\end{equation}` 包起來。數學式會獨立成行，並且擁有一個編號。
- Displaymath 環境，通常直接使用 `\[...\]` 包住數學式即可。數學式獨立成行但無編號。
- Inline math 環境。使用 `$...$` 包住數學式。數學式與普通文字同一行，並且字體一樣大。

下面分別示範這三種顯示模式。

### Equation, new line and numbered

start equation

$$\oint_C E \cdot d\ell = 0 \quad (3.1)$$

end equation

### Displaymath, new line

start equation

$$\oint_C E \cdot d\ell = 0$$

end equation

### Inline

start equation  $\oint_C E \cdot d\ell = 0$  end equation

```
1 \textbf{Equation, new line and numbered}\\
2 start equation
3 \begin{equation}
4 \oint_C {E \cdot d\ell} = 0
5 \end{equation}
6 end equation \\
7 -----\\
8 \textbf{Displaymath, new line} \\
9 start equation
10 \[
11 \oint_C {E \cdot d\ell} = 0
12 \]
13 end equation \\
14 -----\\
15 \textbf{Inline}\\
16 start equation
17 $\oint_C E \cdot d\ell = 0 $
18 end equation
```

以我的習慣來說，一律使用 Equation 環境，即使不一定用到編號。

## 3.8.2 常用數學符號

下面示範常用數學符號。(如果你用的是 L<sup>A</sup>T<sub>E</sub>X 專用編輯器，通常都有按鈕可以輔助插入指令，不用拿個手冊在旁邊對照。) 數學模式中，非指令

表 3.5: 常用數學符號

L <sup>A</sup> T <sub>E</sub> X	Result	說明
<code>A_i=\mathrm{e}^{\{-x\}}</code>	$A_i = e^{-x}$	<code>^_</code> 為上、下標。 <code>\mathrm</code> 標示數學文字
<code>(a+b-c)\cdot d \times e</code>	$(a + b - c) \cdot d \times e$	加、減、乘
<code>\frac{\frac{1}{x}+1/_y}{y-z}</code>	$\frac{\frac{1}{x}+1/y}{y-z}$	兩種除法
<code>\cos^2 \theta - \sin^2 \theta</code>	$\cos^2 \theta - \sin^2 \theta$	三角函數
<code>\alpha\beta\gamma\Omega</code>	$\alpha\beta\gamma\Omega$	希臘字母
<code>50\text{baht}</code>	50B	一隻喬巴 (須要 textcomp 套件)
<code>\infty</code>	$\infty$	富間

文字均視為變數，而變數一律以斜體表達。為了避免一些非變數文字變成

斜體就必須特別處理。(載入 `amsmath` 套件後) `\text` 與 `\mathrm` 就是處理這個問題。`\text` 通常用在純文字上，而 `\mathrm` 則是符號類文字。<sup>23</sup>

### 3.9 科學符號

這兩者雖然有時可用其他方法達成，譬如像溫度  $^{\circ}\text{C}$ ，有人會用圓圈加上 C，有人用字碼。但基於排版以及可讀性我會建議各位用專用套件 ( $\text{\LaTeX}$  style :D)。

#### 3.9.1 SI 單位

使用 `siunitx` 套件。一些常用的如溫度，我會另外寫指令加速完成。如何寫見後面章節說明。

30 cm

5 ml

25  $^{\circ}\text{C}$

37  $^{\circ}\text{C}$  `<- my command`

```
1 $\SI{30}{\centi\metre}$ \\
2 $\SI{5}{\milli\litre}$ \\
3 $\SI{25}{\degreeCelsius}$ \\
4 \temp{37} <-- my command
```

#### 3.9.2 化學式

前面表格已經示範了。使用到 `mhchem` 套件。表達離子帶電用  $^$ ，原子數用  $_$ 。離子帶電可以不加上標，因為用在後面預設視為上標模式。

$\text{Ca}_2^+$	x
$\text{Ca}^{2+}$	o
$\text{Ca}^{2+}$	o

```
1 \begin{table}[h]
2   \centering
3   \begin{tabular}{| 1 | 1 |}
4     \hline
5     \ce{Ca2+} & x \\ \hline
6     \ce{Ca^2+} & o \\ \hline
7     \ce{Ca^{2+}} & o \\ \hline
8   \end{tabular}
9 \end{table}
```

### 3.10 標記與參考 Cross-referencing

「標記與參考」是  $\text{\LaTeX}$  很好用的一個功能。各位千萬不要手打「見圖 1.1」啊!! 標記可以設定一個標籤給它，而內文可以藉由參考指令取得標記位置的數值。這功能可以用在圖、表格、章節、文獻之上。

<sup>23</sup><http://tex.stackexchange.com/questions/19502/is-there-a-preference-of-when-to-use-text-and-mathrm>

表 3.6: 常用標記類型

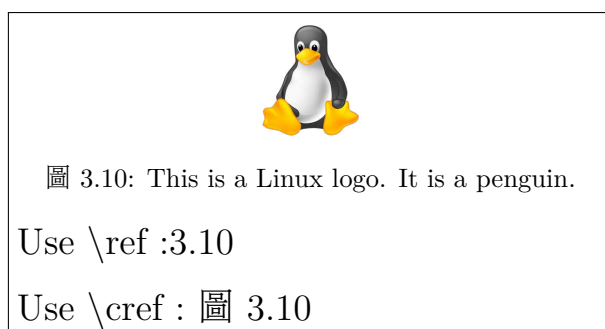
fig	figure
tab	table
eq	equation
sec	section

### 3.10.1 標記 (labal)

標記方法前面已經提到，用`\labal{type:mylabal}` 標記。type 方面如表 3.6所示。有些套件會用到這個縮寫，所以最好符合通用習慣。(其餘見[https://en.wikibooks.org/wiki/LaTeX/Labels\\_and\\_Cross-referencing#Introduction](https://en.wikibooks.org/wiki/LaTeX/Labels_and_Cross-referencing#Introduction)) 稍微注意的是`\labal{}` 要放在`\caption{}` 之後才會有作用。(caption 也不能為空喔!)

### 3.10.2 參考 (ref/cref)

假設你已經在 figure 中使用`\labal{fig:image001}`，那文章中若要產生此圖的標號可以用`\ref{fig:image001}`，但有時我們懶得每個圖都要打“fig. `\ref{fig:image001}`”這樣，所以我這使用了 cleveref 套件。它提供`\cref`指令自動產生相對應的文字。<sup>24</sup>



```

1 \begin{figure}
2   \centering
3   \includegraphics[width=0.2\textwidth]{logo-Linux.png}
4   \caption[Linux logo]{This is a Linux
5     logo. It is a penguin.}
6   \label{fig:reftest}
7 \end{figure}
8 Use \textbackslash{}ref : \ref{fig:
9   reftest}\\
10 Use \textbackslash{}cref : \cref{fig:
11   reftest}

```

## 3.11 文獻

如同之前的`\labal`、`\ref` 插入引用文獻須要建立文獻條目，再利用`\cite{cite_key}` 來插入條目編號。<sup>25</sup>

<sup>24</sup>文字可以由`\crefname` 修改，見 cls 檔。cleveref 亦提供大寫版`\Crefname` 用於行首風格（大寫）。不過中文無此須求。

<sup>25</sup>下面提到的 bibtex, biber, biblatex, natbib 詳細資訊可參考 <http://tex.stackexchange.com/questions/25701/bibtex-vs-biber-and-biblatex-vs-natbib>

### 3.11.1 嵌入式文獻資料庫

少量條目時可以直接使用嵌入的方式，於 thebibliography 環境以 `\bibitem{cite_key}` 嵌入至 tex 檔中，然後以 `\cite{cite_key}` 插入文章。<sup>26</sup>

```

1 \begin{thebibliography}{9}
2
3 \bibitem{lamport94}
4   Leslie Lamport,
5   \emph{\LaTeX\ : a document preparation system},
6   Addison Wesley, Massachusetts,
7   2nd edition,
8   1994.
9
10 \end{thebibliography}
11 see bibliography \cite{lamport94}.
```

“lamport94” 即為 cite\_key 。而 `\begin{thebibliography}{9}` 的 9 指得是生成的條目上限為 9 條。據 wikibook 指出嵌入式上限為 99。

### 3.11.2 BibTeX

**本樣板採用的是 BibLaTeX，請見下節。**

針對大量文獻，通常使用文獻管理軟體先產生一個包含各條目的.bib 檔（亦為純文字檔），再於 tex 中引入。傳統上使用 BibTeX 這軟體，只要在要生成文獻地方插入 `\bibliography{Path_of_BibTeX_database}` 即可。

```

1 \bibliographystyle{plain}
2 \bibliography{sample1,sample2,...,sampleN}
3 % Note the lack of whitespace between the commas and the next bib file.
```

這裡示範引入多個 bib 資料庫，也就是 sampleN，注意中間不能有空格。若要使用與編譯文件不同目錄的資料庫，可以使用相對路徑表達，如 “../sample1”。<sup>27</sup> 一開頭的 `\bibliographystyle` 用途則是指定生成的風格。

用了 BibTeX 之後，記得要再多執行一次編譯動作。也就是說要經過 `xelatex tex` → `bibtex tex` → `xelatex tex` 三個動作。tex 附檔名 (.tex) 不用加。

BibTeX 生成的文獻編號一律是數字，如果你須要更有變化譬作者和年份 (eg. Roberts, 2003)，那你可以使用 Natbib 套件，不過 `\cite` 指令須要修改。這部份請自行查詢 wikibook<sup>28</sup>。

<sup>26</sup>這裡因為示範生成檔會導致章節錯亂，所以只列出 L<sup>A</sup>T<sub>E</sub>X 碼。

<sup>27</sup>tex 檔中的路徑一律使用正斜線 “/”，即使是在 windows 中編譯。

<sup>28</sup>[https://en.wikibooks.org/wiki/LaTeX/Bibliography\\_Management#Natbib](https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#Natbib)



### 3.11.3 BibLaTeX

另一個管理大量文獻的方法則是使用 BibLaTeX。雖然這裡說是「使用 BibLaTeX」，但實際上處理文獻資料庫的是 biber 這個程式。<sup>29</sup> BibLaTeX 只是一個套件，作為 L<sup>A</sup>T<sub>E</sub>X 與 biber 的中間介面，作用如同上面提到的 Natbib 套件。BibLaTeX biber 的好處是支援 UTF-8，能夠處理非英語系的文獻條目。並且資料庫格式新增如 online 類別用來紀錄網頁內容。另外對於文獻的排序方式也有多種選擇。因此本樣板使用 BibLaTeX biber 作為預設的文獻管理方式。（不過 BibLaTeX 畢竟是較新的方法，如果你要發期刊可能不被接受，只能使用 BibTeX。）

BibLaTeX 與 BibTeX 非常類似。首先你要有一個 BibLaTeX 的資料庫，副檔名一樣是.bib，內容格式大同小異。接著使用如下指令：

```
1 \addbibresource{sample1.bib} % must including the extension
2 \addbibresource{sample2.bib}
3 \printbibliography[title = {Bibliography title}, heading = bibnumbered]
```

\addbibresource 作用為加入文獻資料庫。與 BibTeX 的\bibliography 有以下差異：

- 僅加入文獻資料庫而非印在當前位置。
- {} 內僅允許加入單一資料庫，但可多次使用\addbibresource 來加入多個文獻資料庫。
- 文獻資料庫路徑（檔名）須要加上副檔名 (.bib)

加入的資料庫最終會在\printbibliography 的位置生成文獻列表，資料庫有但未用到的文獻不會顯示出。選項中“title”直接指定文獻頁的章標題（中文就是「參考文獻」了）<sup>30</sup>

文獻列表的風格是在引入套件時設定的，下面是樣板 (\*thesis.cls) 內的指令（一般 tex 內用\usepackage 引入套件，cls 內要用\RequirePackage）

```
1 \RequirePackage[backend=biber, natbib=true, style=nature, citestyle=numeric-comp, sorting=none]{biblatex}
```

<sup>29</sup>亦可使用 BibTeX 作為 BibLaTeX 的後端。

<sup>30</sup>我的經驗是 BibLaTeX 不會使用到\bibname 這個變數，所以改了也沒用。

依此設定風格將會是 `nature`，`citestyle` 是數字 “[1]”，且列表順序照文內出現順序。其他風格或是排序方法請自行參考套件手冊並自行於 `cls` 檔中修改。<sup>31</sup>

編譯方面與 `BibTeX` 類似，一樣須要多執行一次編譯動作，只不過中間換成 `biber`。也就是說要經過 `xelatex tex`  $\rightarrow$  `biber tex`  $\rightarrow$  `xelatex tex` 三個動作。`tex` 附檔名 (`.tex`) 不用加。

## 3.12 程式碼

程式碼區域有幾項不同於內文的特性：

- 使用等寬字型
- 忽略  $\text{\LaTeX}$  關鍵字、不分段、保持縮排與換行
- 最好有行號、語法高亮度

建議使用 `listings/listingsutf8` 套件來達成上述目標。（樣板已有）

## 3.13 註解

有時候文章須要做一個小說明，或是寫一些只有編輯者才看得到的註解（無論是寫內心 `os` 還是暫時隱藏文字）。這時候你就須要了解  $\text{\LaTeX}$  註解方法了。

### 3.13.1 $\text{\LaTeX}$ 原生註解

有三種：

- `%` 直接隱藏後面直到行尾。
- 腳註：`\footnote{}`，註解在頁尾，指令位置插入標號。
- 邊註：`\marginpar{}`，註解在頁邊，指令位置不插標號。

使用如下：（ $\text{\LaTeX}$  code 展示套件似乎無法顯示頁邊，所以這裡就不示範結果了。）

---

<sup>31</sup>排序方法還有依作者字母排序，也有相應的 CJK 排序。

```

1 \footnote{This is footnote.}
2 \marginpar{This is marginpar.}

```

### 3.13.2 TodoNotes 套件

顧名思義，TodoNotes 就是給你做便條用的。它可以簡單的做出內文、頁邊註解，並且顯示在生成檔 (pdf) 上。要隱藏註解時只要套件選項改成 disable 即可，如下：

```

1 \usepackage[disable]{todonotes}

```

cls 檔中也有定義了幾個顏色可供使用。如：

這裡示範  
「不確定」

`\todoUnsure{note}`

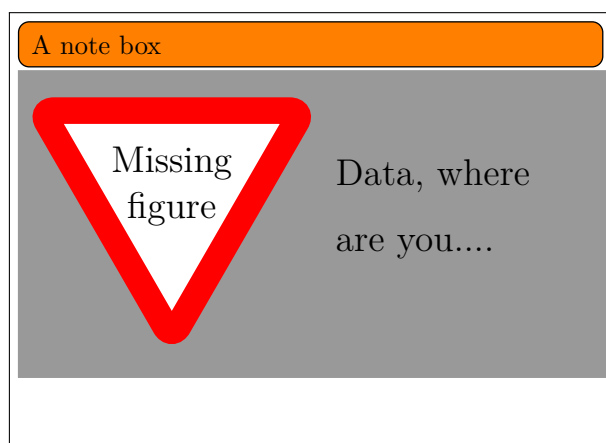
這裡示範  
「要修改」

`\todoChange{note}`

這裡示範  
「資訊說明」

`\todoInfo{note}`

下面這兩個也很好用：



```

1 \todo[inline]{A note box}
2 \missingfigure{Data, where are you....}

```

TodoNotes 搭配上章節 5.1 提到的 SyncTeX 簡直就是神物啊!!! 不過注意用了之後  $\text{\LaTeX}$  要編譯到第二次以後頁邊註解才會指向正確位置。

為了讓各位覺得使用本樣板更值回票價，我在 config.tex 中放置了 publish 選項。

```

1 \setboolean{publish}{false} % {true}/{false}

```

只要你設定 true 之後將自動隱藏 TodoNotes。



## 四、 進一步了解

這裡放一些進階知識。

### 4.1 分割文件

如同寫程式般，大型文件（論文就算是啦！）通常不會一個檔案完成，而是依章節分割成多個子檔撰寫後再以一個主檔合併。以本樣板及教學為例，主檔都是 `main.tex`。<sup>1</sup> `main.tex` 中再以 `\input{filename}` 引入其他 `tex` 檔。除 `\input{}` 外，亦可用 `\include{}`。兩者差異在於 `\input{}` 一定會另起一頁，且雙面配置時一定由右頁開始<sup>2</sup>。這兩個引入的指令不須另加套件即可使用<sup>3</sup>。

若要做到可單獨編譯子檔（節省編譯時間），需要使用 `subfile` 或是 `standalone` 套件。（本樣板預設採用 `standalone` 套件。所須的子檔標頭已經寫在 `chapter_template.tex` 中，直接複製進行修改即可。）

### 4.2 字型

電腦上常用的中文字型有四大基本類型，分別為楷、明、宋、黑（這是不包含行書之類的非正式文書字體）。楷體較粗，明、宋較細。而黑體則是各筆劃粗細相同（常用於廣告？）。一般來說文件的內文應該要是較細的文字（如明、宋體），避免閱讀疲勞。因此此樣板設定僅封面為楷體，其餘皆為明體。<sup>4</sup>

除了字的美觀外，各位可能要注意一下字本身的正確性。因為有些字體使用舊的寫法或是大陸寫法，如「骨」字上面應該是向右而非向左。

---

<sup>1</sup>檔名隨你訂，不限於 `main.tex`。

<sup>2</sup>預設起始頁為右頁，可自訂為左頁。

<sup>3</sup>ShareLaTeX 網站提到用到巢狀引用時用 `import` 套件較好

<sup>4</sup>對應到微軟預設字型就是標楷體以及細明體。新細明體與細明體差別為英文字元是否等距。

關於中文的開源字型可見 Cheng-Chia Tseng 的詳細講解。<http://breezymove.blogspot.tw/2013/11/cantarell.html>。而  $\text{\LaTeX}$  的搭配可以再參考「阿盤」的設定 (<http://www.slideshare.net/chenpanliao/apan-beamertemplate>)。各字型由 Google 搜尋字型名稱就可以找到下載點了。除了上面網頁提到的字型外，政府網站也提供「全字庫」字型。<sup>5</sup>。在意字體美觀的人推薦使用  $\text{cwTeX}$  的字型（Debian 套件庫已經有了，不過是舊版本。新版請 google）。<sup>6</sup> 英文字型的分類就簡單的多。一般就分兩大類：襯線與無襯線。Windows 上最常見的 Times New Roman 就是襯線字型。一般來說內文的英文用襯線會比較美觀。另外英文還有一種等寬字型。這是因為英文字母若均為相同寬度，一般文件會顯得太零散導致不易閱讀，因此一般使用非等寬的字型（如 `llll` 與 `wwwwww` 的寬度是不同的）。但在如程式碼這種須要對齊字元的用途上，等寬字元則是比較適合的字型。

此樣版可以在 `config.tex` 中分別設定中文以及英文字型。

### 4.3 自製指令與巨集

有的時候可能會覺得在寫文件的時候，一些東西很常用，但每次打都很麻煩，譬如之前的溫度符號“ $^{\circ}\text{C}$ ”。或是非英語、超長文字.....。這時候可以善用  $\text{\LaTeX}$  的程式化特性。定義新的指令可以用：

```
1 \newcommand{name}[num]{definition}
```

“name”就是指令名稱。“[num]”是參數數目。`{}` 內就是定義啦！直接打就可以顯示出來了。我們直接看範例：

```
Someone:      Poincarè ,    Does
Schrödinger's cat get pneumonoul-
tramicroscopicsilicovolcanoconiosis?
Poincarè: I just care my three-body
problem.
Cat : Here is soooo cold , -195 $^{\circ}\text{C}$  ,
let me out !
```

```
1 \providecommand{\catman}{Schr\`{o}dinger}
2 \providecommand{\tbpman}{Poincar\`{e}}
3 \providecommand{\pne}{
  pneumonoultramicroscopicsilicovolcanoconiosis
}
4 \providecommand{\temp}[1]{\SI{#1}{\
  degreeCelsius}}$}
5 % -----
6 Someone: \tbpman{} , Does \catman \
  \textquoteright{s cat get \pne ? \
7 \tbpman : I just care my three-body problem
  . \
8 Cat : Here is soooo cold , \temp{-195} ,
  let me out !
```

<sup>5</sup>抱怨一下，這東西用納稅人的錢做的，結果只是能免費使用，並不是開放字型....

<sup>6</sup>曾見過有人認為 Linux 中常用的「AR PL UMing TW」以及對應的楷體很醜，所以替換成全字庫字型。

這個範例使用的是`\providecommand`，因為`\newcommand`遇到已經定義過的名稱會出錯（這其實是為了檢查問題），而`\temp`已經定義過了，所以這裡使用`\providecommand`來跳過避免重複定義。類似的還有`\renewcommand`，剛好是介於前面兩者之間，遇到沒定義過的會出錯。

定義指令也可以使用 Plain  $\text{T}_{\text{E}}\text{X}$  的`\def`，它同樣也可以接受輸入參數，`config.tex` 內的設定大多由`\def`完成，一方面是它不檢查重複定義（不同於`\providecommand`忽略第二次定義，而是重新定義，覆蓋原有設定<sup>◦</sup>），另一方面是字母少，美觀:D。

對於變動數量的指令可以使用 `xparse` 或 `xargs` 來處理。這裡推薦使用 `xparse`，他是  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  3 計畫中用來替換  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  2<sub>ε</sub> 的`\newcommand`等指令。據說它對於參數省略的狀況處理得比 `xargs` 好。<sup>7</sup> `xparse` 的指令集和原始的定義指令很像，分別是（這裡只列 3 個）：

- `\NewDocumentCommand`
- `\RenewDocumentCommand`
- `\ProvideDocumentCommand`

而在指令名稱後不是以 “[num]” 單純定義個數，取而代之的是設定各參數的屬性。直接看我的 `cls` 內定義快速插圖的  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  Code:

```

1 % easy insert single figure
2 % \insfig[scalefactor][label][!htb]{path}[caption][short caption]
3 \NewDocumentCommand\insfig{0{0.5} o 0{} m 0{} o}{ % need xparse package
4   \begin{figure}[#3]%[!htb]
5     \centering
6     \includegraphics[width=#1\textwidth]{#4}
7     \IfNoValueTF{#6}{\caption{#5}}{\caption{#6}{#5}}
8     \label{\IfNoValueTF{#2}{fig:#4}{#2}}
9   \end{figure}
10 }
```

其中 “`{0{0.5}} o 0{} m 0{} o`” 就是定義最多 6 個參數，大寫 O 表示那個參數可省略，預設值為 `{}` 內容。小寫 o 同樣可略，但會傳回 “-NoValue-” 標記，這裡就是用 `\IfNoValueTF` 來判斷是否省略。m 表示必要參數。使用時只要是可略參數都要用 “[ ]”，必要參數則是 “{ }”。除了 o,m 之外，`xparse` 還提供其他不同的參數與指令，請自行查閱說明手冊。

<sup>7</sup>本樣板兩個都有用，因為 `todonotes` 的顏色設定是複製別人的。而他用的就是 `xargs`，懶得自己再改 XD。

## 4.4 文件類型 (documentclass) 選項

L<sup>A</sup>T<sub>E</sub>X 文件的第一行指定了使用何種類型的文件。如同一般指令，它可以接受輸入選項。以 book class 來說，最常用到的選項如下：

```
1 \documentclass[a4paper,10pt]{book}
```

a4paper 指定使用 A4 紙，10pt 指定文件基準字體大小為 10pt<sup>8</sup>。其他還有很多選項，一般依照預設值就可以了。不過有一個選項“draft”很有用。它會略去圖形、連結的處理。所以編譯速度會較快 (以這份教學為例，draft 約減少 20% 時間)。要啓用的話只要將它加入方括號中即可。本樣版採用 standalone 處理子文件，這裡分別示範主文件、子文件添加選項方法。

主文件：

```
1 \documentclass[draft]{NCU_thesis}
```

子文件：

```
1 \documentclass[class=NCU_thesis, crop=false, float=true, draft]{standalone}
```

子文件僅影響編譯子文件時的效果，而主文件影響整體。

## 4.5 tex 檔編碼

建立 tex 檔時請注意使用的文字編碼是否為 UTF-8，通常流行的 L<sup>A</sup>T<sub>E</sub>X 編輯器均預設 UTF-8。為了避免編譯時產生問題，建議非純英文文件均使用 UTF-8 編碼。

## 4.6 樣板提供工具

這裡列一些我論文裡面用到的小工具。

### 4.6.1 圓圈文字

有時後會用到一些記號，像是提到公司、產品時常會用：

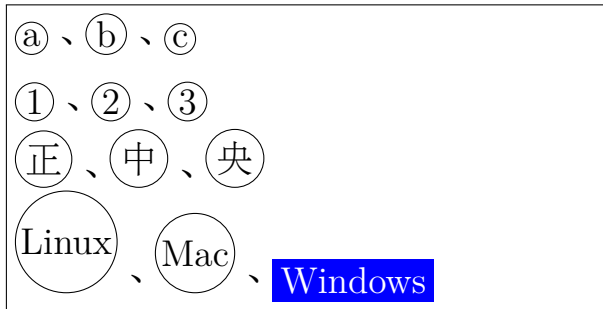
©,®,™

```
1 \textcopyright,\textregistered,\  
texttrademark
```

<sup>8</sup>book class 僅接受 10pt,11pt,12pt, 12pt 對中文來說仍略小，因此本樣版採用\fontsize 進行無級調整，所以樣板的 cls 中雖然設定 10pt，但文件字體仍為\fontsize 設定值。唯一被影響到的是自動填單區域。由於我一開始使用的是預設值 10pt 進行文字定位，加上\fontsize 不知為何字體大小不起作用（但行距仍作用），所以 cls 中維持 10pt 設定。



但假如要更多樣的符號怎辦？沒問題。本樣板提供



```
1 \circhar{a}、\circhar{b}、\circhar{c}\\
2 \circhar{1}、\circhar{2}、\circhar{3}\\
3 \circhar{正}、\circhar{中}、\circhar{央}\\
4 \circhar{Linux}、\circhar{Mac}、\colorbox{
  blue}{\color{white}Windows}
```

(咦!Windows 為何沒有成功？啊！BSOD 了....)

#### 4.6.2 快速插入單一圖片

之前提過，見圖片的章節 (章節 3.5)。

#### 4.6.3 單位

提供一些單位，指令名稱不喜歡可以自改，位於 `macros_document.tex` 中。要更多的單位請參考 `siunitx` 套件的手冊。



```
1 \temp{37}\\
2 \symumm{300}=\SI{30}{\centi\metre}\\
3 \symumv{10}\\
```

#### 四、進一步了解

## 五、寫作輔助工具

### 5.1 SyncTeX

由於 L<sup>A</sup>T<sub>E</sub>X 非所見即所得的排版軟體，如果我們編譯後發現須修改生成檔 (dvi/pdf) 的特定位置，往往要辛苦找尋原始碼中的對應位置。SyncTeX 即是一個解決這個麻煩的方法。他可以讓你點閱讀器後，自動將編輯器移到對應位置。使用的方法是在編譯時加入 “`--synctex=1`” 的選項 (或是 `-synctex=1`)，或是在 tex 檔中插入 “`\synctex=1`”。(除非編譯器太舊，不然都支援。) 本樣板已於 config.tex 中設定了 `\synctex=1`。

#### 5.1.1 Kile + Okuar

我用的寫作環境是 Kile 編輯器 + Okuar 檢視器。只要將 Kile 檢視 pdf 的動作改用 “ForwardPDF”，然後在 Okuar 中設定 Editor 為 Kile 這樣就可以了。只要在 Okuar 瀏覽 pdf 時按下 *shift* + 滑鼠左鍵，Kile 就會自動跳過去了。這個網頁有詳細圖文教學：<https://localframeofreference.wordpress.com/2010/08/31/kile-synctex/>

### 5.2 Git 版本管理與 tex 檔比較

其實這是兩個不同主題的東西，不過我不打算在這裡細講，只概略提一下。

Git 是一個版本管理系統，最初用來管理程式碼 (Linux kernel)。不過由於他的特性，使得他可以被用來管理任何形式的檔案。應用於論文上除了能管理各個版本之外，更棒的是備份容易。搭配 latexdiff 這個比較 tex 檔版本的工具後，你還可以清楚知道各版本的變動狀況。我之前有做一個投影片有較詳細的說明，請見：<http://www.slideshare.net/sppmg/latex->

with-git 而遠端 git 倉庫可以使用 “GitLab” 這個免費的服務，他可以讓你選擇是否隱藏專案（也就是你的論文啦！）。

論文這種東西隨時隨地多備份比較妥當（當然還有相關資料）。當年中央被偷筆電隨身碟的博士生不知最後如何了..... 之前還有清大研究鍺烯的博士生被撞，焚毀筆電事件。

### 5.2.1 gitignore

稍微說明一下 gitignore 這個檔案。Git 可以將某目錄下所有檔案加入追蹤。但有些檔案是我們不要的暫存檔或備份檔，如 L<sup>A</sup>T<sub>E</sub>X 編譯產生的 \*.aux、\*.log 或是編輯器自動備份的 \*.backup 等等。所以我們可以在 git 根目錄加上 “.gitignore” 這個檔案，這樣在大範圍加入檔案時就會略去符合條件的檔案了（仍然可以用 “-f” 強制加入，如最終生成的 pdf）。<sup>1</sup>。L<sup>A</sup>T<sub>E</sub>X 用的.gitignore 內容為：

```
*.aux
*.glo
*.idx
*.log
*.toc
*.ist
*.acn
*.acr
*.alg
*.bbl
*.blg
*.dvi
*.glg
*.gls
*.ilg
*.ind
*.lof
*.lot
*.maf
*.mtc
*.mtc1
*.out
*.synctex.gz

# sppmg. for editor
*.bcf
*.backup
*~
*.kilepr

# sppmg. this only
*.xml
*-blx.bib
*.pdf
```

<sup>1</sup>這是 Linux 的檔名，加. 表示隱藏檔。其他系統請自行查詢。

```
*.sta  
*.directory  
*_latexmk  
.*
```

## 五、寫作輔助工具

## 六、 文獻搜尋

文獻除了直接由 google 搜尋關鍵字外，亦可由專門的文獻資料庫搜尋。(付費型) 文獻資料庫可以詢問各校圖書館 (網頁上通常會寫明)，下面幾個是我推薦的搜尋網站。

**ScienceDirect** <http://www.sciencedirect.com> 免費使用，但全文下載仍須權限。

**PubMed.gov** <https://www.ncbi.nlm.nih.gov/pubmed> 免費使用，但全文下載仍須權限。

**Web of Science** <http://www.webofknowledge.com> 須權限才能使用。文獻資訊包含引用次數等等的進階資訊。

「權限」指的就是須要付費才能取得。通常學校或研究單位付費購買後，同一個網域的使用者都能用。而網域之外的使用者就要看單位是否提供校外連線方法。以 NCU 來說，校外使用者可以使用 EZproxy。我的用法是直接改網址，<https://ezproxy.lib.ncu.edu.tw/login?url=<url>>。[<url>](https://ezproxy.lib.ncu.edu.tw/login?url=<url>) 表示完整(含 <http://>)的網址。所以<http://www.sciencedirect.com> 要載付費全文的話就是由<https://ezproxy.lib.ncu.edu.tw/login?url=http://www.sciencedirect.com> 登入後下載。至於快速改書籤的方式在 Firefox 中可以由建立關鍵字書籤方式處理。如下圖：於網址填入<http://ezproxy.lib.ncu.edu.tw/login?url=http://%S> 並加上關鍵字 ezp，當瀏覽到須要登入的頁面時直接在網址開頭加上 “ezp ” 即可轉換成所須網址。<sup>1</sup>

<sup>1</sup>同理，NCU 圖書館查書可以用[http://opac.lib.ncu.edu.tw/search\\*cht/a?a?searchtype=Y&searcharg=%S](http://opac.lib.ncu.edu.tw/search*cht/a?a?searchtype=Y&searcharg=%S) 以及關鍵字 “lib” 來處理。

## 六、文獻搜尋

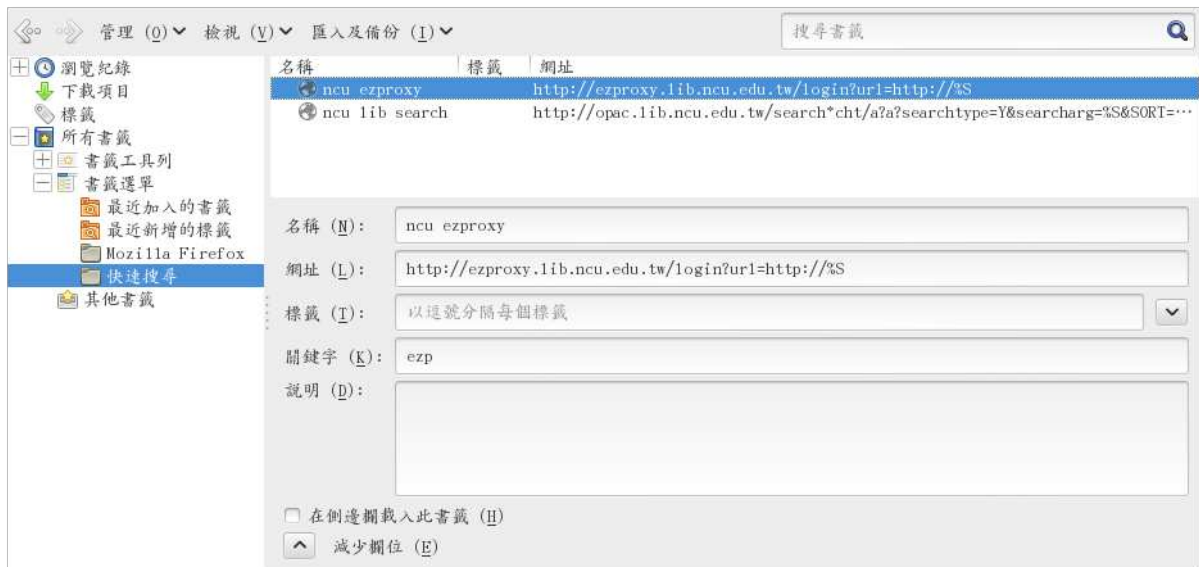


圖 6.1

除了上面列的網站外，這裡特別推薦 arXiv.org(<https://arxiv.org/>)，台大圖書館有詳細介紹 (<http://tul.blog.ntu.edu.tw/archives/17366>)。



## 七、 特定樣板細說

### 7.1 中央大學 (NCU)

#### 7.1.1 章節層級

book 類別的章節層級有：

1. `\chapter{章}`
2. `\section{節}`
3. `\subsection{小節}`
4. `\subsubsection{小小節}`
5. `\paragraph{段}`
6. `\subparagraph{小段}`

不過學校體例參考文件中少去「`\subsubsection{小小節}`」。在樣板中將以`\subsubsection{小小節}`來代替段的地位。對於`\subsubsection`以下層級，學校要求修改標題的標號，並且整區進行縮排。我自己大多使用到第 2 級 (1.1.1)，第 3 級 (1.1.1.1) 只有用到一次，所以第 3 級以後的樣式我直接照 L<sup>A</sup>T<sub>E</sub>X 預設值。不過本樣板還是提供了較為符合格式的風格選項給各位。

##### 7.1.1.1 目錄風格

樣本預設目錄僅顯示到第 2 級 (`\subsection{小小節}`)，要調整的話可以在 `config.tex` 中修改 `tocdepth` 的數值。而目錄風格（各章節等級的縮排策略）方面，本樣板提供兩種選擇，可經由 `config.tex` 中的 `tocStyle` 切換，其結果如圖 7.1（紅框是 pdf viewer 標示連結，不會印出）。`\tocStyle{0}` 表

一、	aaa.....	一、	aaa.....
1.1	aaa.....	1.1	aaa.....
1.1.1	aaa.....	1.1.1	aaa.....
二、	aaa.....	二、	aaa.....
2.1	aaa.....	2.1	aaa.....
2.1.1	aaa.....	2.1.1	aaa.....

(a) `\def\tocStyle{0}`                      (b) `\def\tocStyle{1}`

圖 7.1: 樣板提供的目錄風格

示幾乎按照  $\text{\LaTeX}$  的方式縮排，`\tocStyle{0}` 表示完全符合學校規定縮排（不過章數值會距離過遠，我認為較醜）。

#### 7.1.1.2 標題風格

標題風格提供三種形式讓各位自行選擇要採用哪一種。（下面的示範為避免影響到主文件章節數值，使用圖片展示編譯結果。）

首先示範使用  $\text{\LaTeX}$  預設值（`\styletitles{0}`）的情況（這裡使用巨集方式實現功能，實用上不須要在同一份文件中改變，因此樣本以變數形式設定風格以及縮排距離），如圖 7.2:

接著是修改標題形式成學校樣式，但不進行縮排。如圖 7.3

最後是連縮排也符合學校規定（長度自訂）（圖 7.4）。不過內文同時縮排須要調整`\leftskip`的數值，目前樣板須要手動於縮排前後修改。並且記得以空行分隔上一個段落，否則無法產生效果。`\leftskip`只有在碰到`\subsubsection`層級之後才須要加，碰到的機會應該不多。



圖 7.2: `\styletitles{0}`

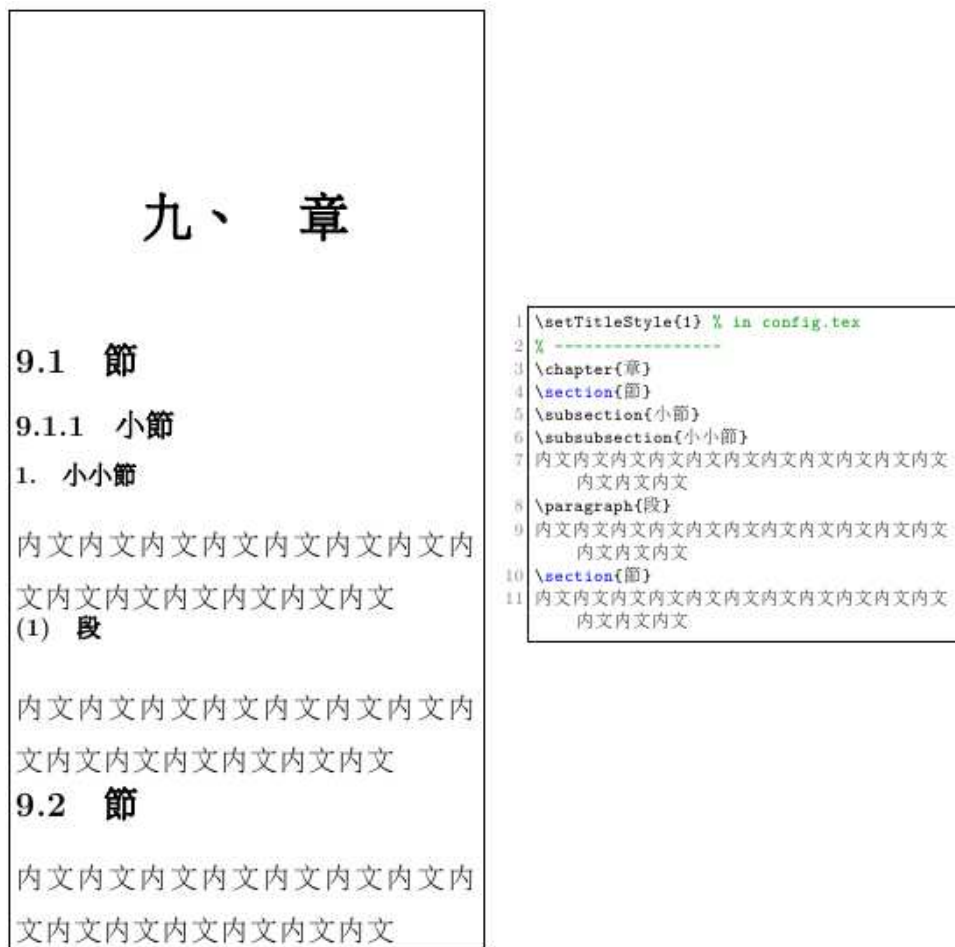


圖 7.3: `\styletitles{1}`

十、章

10.1 節

10.1.1 小節

1. 小小節

内文内文内文内文内文内文内文  
内文内文内文内文内文内文内文  
内文  
(1) 段

内文内文内文内文内文  
文内文内文内文内文内文  
内文内文内文内文内文  
文

10.2 節

内文内文内文内文内文内文内文  
文内文内文内文内文内文内文

```
1 \setTitleStyle{1}[1cm][3cm] % in config.tex
2 % -----
3 \chapter{章}
4 \section{節}
5 \subsection{小節}
6
7 \setlength{\leftskip}{\indentblocksss} %
8   indent for subsubsection
9 \subsubsection{小小節}
10 内文内文内文内文内文内文内文内文内文内文内文
11 内文内文内文
12 \setlength{\leftskip}{\indentblockpar} %
13   indent for paragraph
14 \paragraph{段}
15 内文内文内文内文内文内文内文内文内文内文内文
16 内文内文内文
17 \setlength{\leftskip}{0cm} % recover (no
18   indent)
19 \section{節}
20 内文内文内文内文内文内文内文内文内文内文内文
21 内文内文内文
```

圖 7.4: \styletitles{1} 與\leftskip

## 七、 特定様板細説

## 八、 樣板測試結果及已知問題

### 8.1 測試結果

TeX 發行版	作業系統	XeTeX 版本	測試結果
TeX Live 2012 (2012.20120611-5)	Debian GNU/Linux wheezy(7.9)	3.1415926-2.4-0.9998	章節標題後立即接圖片會導致整段粗體，插入任一行文字可解決。 <code>\url</code> 跨頁會導致頁眉及中間浮動環境均含連結
TeX Live 2016 (2016.20160819-2)	Debian GNU/Linux stretch/sid(9)	3.14159265-2.6-0.99996	無問題
MiKTeX 2016 (???)	Window 7	???	???

### 8.2 待解決問題

- 過長的連結會超出頁邊。本樣板使用 `url` 套件處理這個問題，但仍無法處理超長單字。目前 `breakurl` 套件 (1.40+) 提供了 `anythingbreaks` 選項，可以任意截斷連結，請參考<http://tex.stackexchange.com/questions/107507/biblatex-url-breaking-not-working-in-dvi-mode> 的說明。不過 `breakurl` 僅支援原始的  $\text{\LaTeX}$ ， $\text{\PDFLaTeX}$ / $\text{\XeLaTeX}$  不支援。

## 八、 樣板測試結果及已知問題