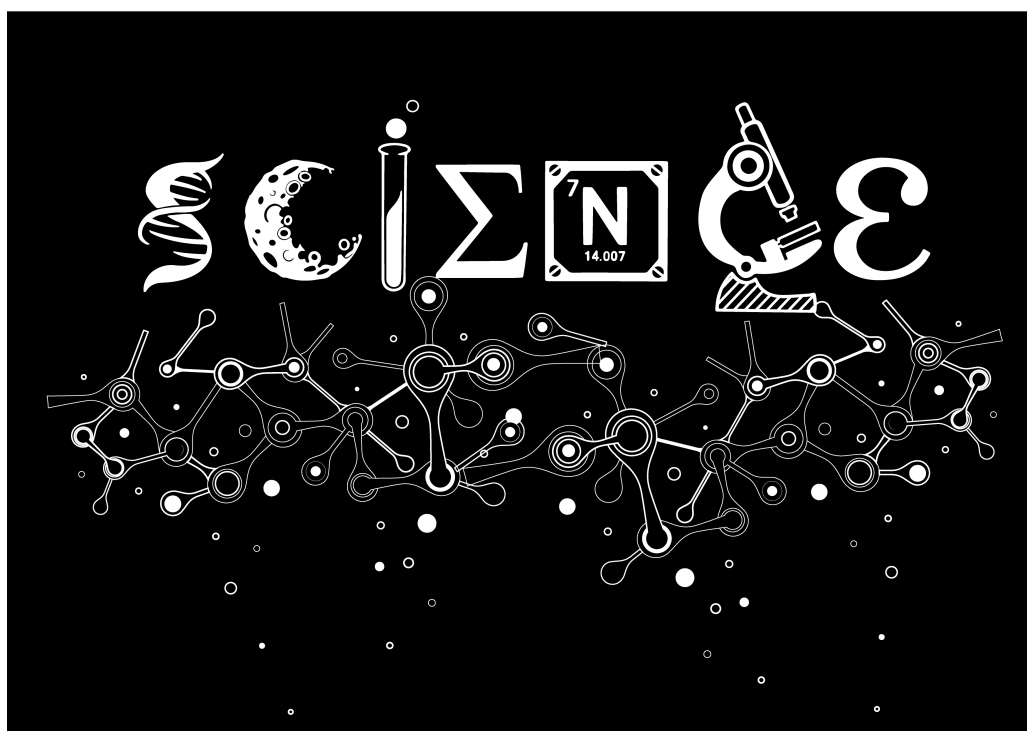


L^AT_EX 祕笈

武陵高中資訊讀書會
武陵高中第十屆科學班

GPwaob_92679



目錄 Table Of Contents

I 前言	
Preface	3
作者介紹	4
推薦序	5
1 L^AT_EX 簡介	6
1.1 什麼是 T _E X ?	6
1.2 什麼是 L ^A T _E X ?	6
1.3 為什麼厲害的人都在用 L ^A T _E X ?	7
1.4 L ^A T _E X 究竟有多強大 ?	8
1.5 L ^A T _E X 與其他軟體的比較	11
2 初始環境設定	13
2.1 L ^A T _E X 的運作模式與編譯流程	13
2.2 安裝 L ^A T _E X 發行版 — MiK _T E _X	14
2.3 安裝套件 — xeCJK	15
2.4 安裝 L ^A T _E X 編輯器 — Texmaker	16
2.5 設定 Texmaker	17
II L^AT_EX 入門篇	
L^AT_EX for Beginners	18
1 寫出你的第一份 L^AT_EX 文件 !	19
1.1 Hello, World! in L ^A T _E X	19

III	\LaTeX 進階篇	
	\LaTeX for Intermediates	20
IV	\LaTeX 專精篇	
	\LaTeX for Experts	21

Part I

前言

Preface

作者介紹

推薦序

第 1 章

L^AT_EX 簡介

歡迎各位讀者進入 L^AT_EX 的世界！在真正開始使用 L^AT_EX 前，先讓我們揭開 L^AT_EX 複雜的身世背景、看看 L^AT_EX 強大的威力吧！（怎麼寫得有點中二 ==）

1.1 什麼是 T_EX？

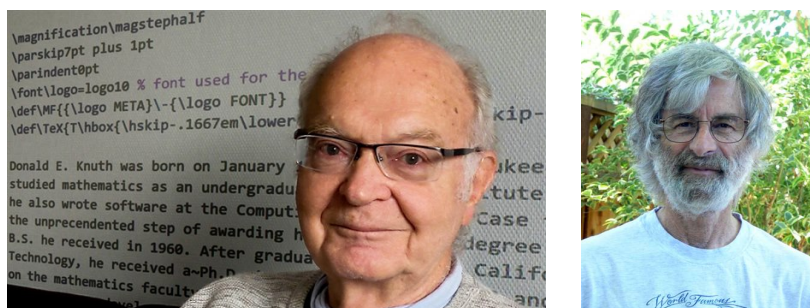
T_EX 是美國電腦科學家 Donald Knuth 最初在 1978 年發表的一套排版軟體，同時也是一種標記式語言 (Markup Language)。相較於市面上大多的排版軟體 (如: Microsoft Word、LibreOffice Writer、Google Docs)，T_EX 沒有漂亮的圖形化介面 (GUI)，而是像寫程式一樣，先把指令 (告訴電腦文字與版面應該長怎樣) 與文件內容 (真的給人看的東西) 寫在一個純文字檔後，再經過編譯器 (Compiler) 的編譯，產生最後可供人類閱讀的文件檔。

※ 為了與 T_EX 後續衍生出的一大堆程式區別，這種最早出現、最陽春的 T_EX 也經常被稱為原版 T_EX (Plain T_EX)。

1.2 什麼是 L^AT_EX？

隨著科技的發展，當年的原版 T_EX 所提供的功能早已不敷使用，同時也被人覺得太複雜、不親民。因此，美國又有一位電腦科學家 Leslie Lamport 在 1984 年發表了基於 T_EX 的排版系統 L^AT_EX，支援更多實用的功能和更親民的指令集，也推廣了這整套系統的應用。

講白話一點，L^AT_EX 其實就是 T_EX 的 PRO 版，而且比原本的 T_EX 好用 n 百倍，導致 T_EX 已經被大眾打入冷宮了。

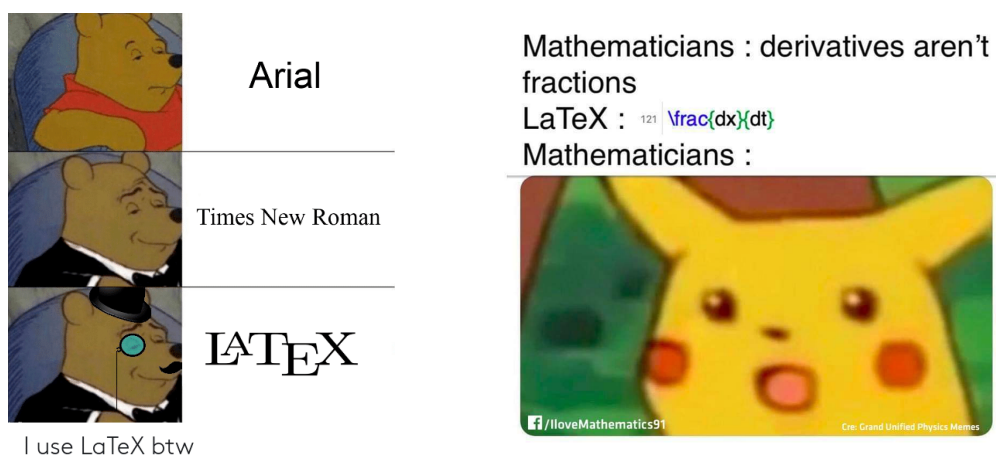


▲ 圖 1.1: (左) Donald Knuth (右) Leslie Lamport

1.3 為什麼厲害的人都在用 L^AT_EX ?

L^AT_EX 的功能非常強大、應用也十分廣泛，除了最主要被用於國際上的各大科學領域的文獻與教科書外，其實我們也可用 L^AT_EX 輕而易舉地做出精美的筆記、報告、履歷、甚至是投影片。以下是幾個 L^AT_EX 的特色與優點：

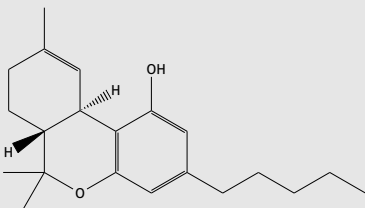
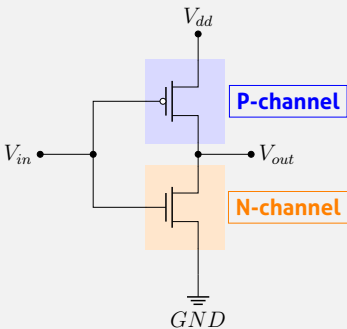
- **數學公式間距調整**：L^AT_EX 幾乎就是為了排版數學公式而生的，其內部的演算法可配合當前字型，自動調整數學公式中數字與符號間的距離；使用者也可用指令手動增加或減少間距大小。
- **萬用的純文字檔案**：L^AT_EX 的原始碼以純文字檔案儲存，因此檢視與編輯時不受作業系統或裝置限制，只要是打得出字的機器都能直接編輯。假設你的檔案是純英文，甚至可以用摩斯電碼傳給你朋友。
- **一勞永逸的格式設定**：一份 L^AT_EX 文件的所有格式設定都是透過指令達成的，因此格式設定只須寫一次，之後的新文件若要使用同樣的格式，就只要無厘頭的把整串格式設定的指令複製過去即可。網路上也有許多已經刻好的範本，可以直接下載下來使用。
- **穩定性**：一樣的原始碼即使在不同機器上編譯，理論上也會輸出一模一樣的檔案。不像 Word 或 Google Docs，有時會出現同個檔案在另一台電腦上就莫名其妙格式跑掉或根本無法開啟的狀況。
- **完全免費**：所有 T_EX 大家族的軟體都是完全免費的，當然也包括 L^AT_EX 與大部分的實用套件 (Package)。
- **廣大的套件支援**：L^AT_EX 目前可支援的套件高達 6,000 多個，幾乎可包辦所有你想得到的需求。裡面除了有方便管理參考文獻、製作圖表與投影片的工具外，其實也藏著許多令人意想不到的套件...。(詳見表 1.1 與表 1.2)
- **神秘感**：讓你看起來很像某個資訊電神在寫程式；或是像某個駭客在入侵學校系統盜段考考卷出來，洩題給同學之後再跟他們討錢。



▲ 圖 1.2: 兩張關於 L^AT_EX 的梗圖

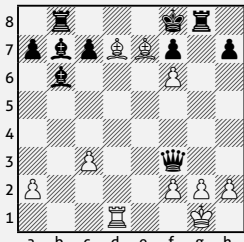
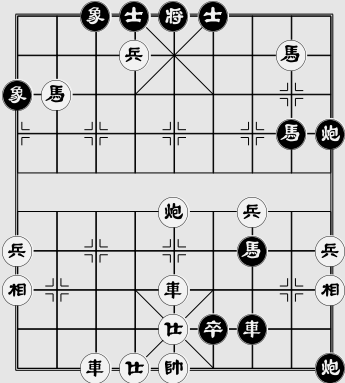
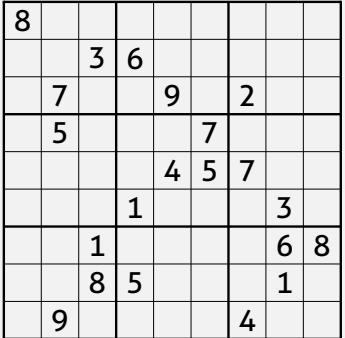
1.4 L^AT_EX 究竟有多強大？

說了這麼多，就讓我們實際看看 L^AT_EX 在數學公式與科學圖表優秀的排版能力吧！

應用	套件	範例	原始碼 (部分省略)
數學公式	amsmath	$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx$	<pre>\[\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx \]</pre>
化學結構式	chemfig	 <p>四氫大麻酚 (Tetrahydrocannabinol, THC)</p>	<pre>\chemname[3ex]{\chemfig[angle increment=30]{ *6(-=(-[1]-[1]-[1]-[1])=-(OH)-(*6(-<[:1]H) (*6(-(-)-)))-(<[7]H)-(-[6])(-[8])-O-))=)}} }{四氫大麻酚 (Tetrahydrocannabinol, THC)}</pre>
電路圖	circuitikz		<pre>\begin{circuitikz} % 繪製彩色方塊與文字標示 \fill[blue!15!white] (-1, 0.8) rectangle -> (0.5,-0.8); \fill[orange!20!white] (-1, -1.2) rectangle -> (0.5,-2.8); \node[draw, color=blue] at (1.7, -> 0){\textbf{P-channel}}; \node[draw, color=orange] at (1.7, -> -2){\textbf{N-channel}}; % 繪製 PMOS 與 NMOS \draw (0,0) node[pmos](P){}; \draw (0, -2) node[nmos](N){}; % 繪製電線、接點與接點文字標示 \draw (P.D) -- (N.D); \draw (P.S) to[short, -*] ++(0, 0.5) -> node[above]{\$V_{dd}\$}; \draw (N.S) -- ++(0, -0.5) node[ground](GND){} (GND.south) node[below]{\$GND\$}; \draw (P.G) -- ++(-1, 0) -- ++(0, -1) -> node[(in)]{} -- ++(0, -1) -- (N.G); \draw (in.center) to[short, -*] ++(-1, 0) -> node[left]{\$V_{in}\$}; \draw (0, -1) to[short, -*] ++(1, 0) -> node[right]{\$V_{out}\$}; \end{circuitikz}</pre>

▲ 表 1.1: L^AT_EX 的科學應用範例

但是這麼強大的軟體，不拿來做一些趣味應用真是太可惜了！其實 L^AT_EX 中也包含許多意想不到的套件，讓我們可以排版出科學用途之外的東西。以下是幾個有趣的例子：

應用	套件	範例	原始碼 (部分省略)
西洋棋	skak texmate	<p>□ Adolf Anderssen Berlin/Berlin GER/1852 ■ Jean Dufresne Evans Gambit – C52</p> <p>1.e4 e5 2.♗f3 ♘c6 3.♗c4 ♗c5 4.b4 ♗×b4 5.c3 ♗a5 6.d4 exd4 7.0-0 d3 8.♖b3 ♗f6 9.e5 ♗g6 10.♞e1 ♗ge7 11.♗a3 b5 12.♖×b5 ♞b8 13.♖a4 ♗b6 14.♗bd2 ♗b7 15.♗e4 ♖f5 16.♗×d3 ♖h5 17.♗f6+ g×f6 18.exf6 ♞g8 19.♞ad1 ♖×f3 20.♞×e7+ ♗×e7 21.♖×d7+ ♗×d7 22.♗f5+ ♗e8 23.♗d7+ ♗f8 24.♗×e7# 1-0</p> <p>Evergreen Game</p>  <p>(Final position after 24.Bxe7#)</p>	<pre>% 繪製標題 \whitename{Adolf Anderssen} \blackname{Jean Dufresne} \chessevent{Berlin/Berlin GER/1852} \chessopening{Evans Gambit} \ECO{C52} \makegametitle % 列印棋譜 \begin{texmate} 1.e4 e5 2.Nf3 Nc6 3.Bc4 Bc5 4.b4 Bxb4 5.c3 Ba5 ↪ 6.d4 exd4 7.0-0 d3 8.Qb3 Qf6 9.e5 Qg6 10.Re1 ↪ Nge7 11.Ba3 b5 12.Qxb5 Rb8 13.Qa4 Bb6 14.Nbd2 ↪ Bb7 15.Ne4 Qf5 16.Bxd3 Qh5 17.Nf6+ gxf6 ↪ 18.exf6 Rg8 19.Rad1 Qxf3 20.Rxe7+ Nxe7 ↪ 21.Qxd7+ Kxd7 22.Bf5+ Ke8 23.Bd7+ Kf8 ↪ 24.Bxe7\# \result{1-0} \end{texmate} % 繪製盤面 \smallboard \notationon \preparediagram{Evergreen Game}{(Final position ↪ after 24.Bxe7#)} \makediagrams</pre>
象棋	cchess		<pre>\begin{position} \piece{c}{10}{B} \piece{d}{10}{G} ↪ \piece{e}{10}{K} \piece{f}{10}{G} \piece{d}{9}{p} \piece{h}{9}{n} \piece{a}{8}{B} \piece{b}{8}{n} \piece{h}{7}{N} \piece{i}{7}{C} \piece{e}{5}{c} \piece{g}{5}{p} \piece{a}{4}{p} \piece{g}{4}{N} \piece{i}{4}{p} \piece{a}{3}{b} \piece{e}{3}{r} \piece{i}{3}{b} \piece{e}{2}{g} \piece{f}{2}{P} \piece{g}{2}{R} \piece{c}{1}{r} \piece{d}{1}{g} \piece{e}{1}{k} ↪ \piece{i}{1}{C} \end{position}</pre>
數獨	sudoku		<pre>\begin{sudoku-block} 8 3 6 7 9 2 5 7 4 5 7 1 3 1 6 8 8 5 1 9 4 \end{sudoku-block}</pre>

五線譜	musixtex		<pre>\begin{music} \nostartrule \nobarnumbers \generalmeter{\meterfrac{4}{4}} % 拍號: 4/4 拍 \generalsignature{+3} % 調號: 3 個升 \startextract % 開始記譜 \Notes{\qa{1}\en\Notes\ibl0n{-1}\qb0{o}\itieu0n ↪ \tqb0n\ibl0n2\ttie0\qb0{n}\itieu0o\qb0{o} ↪ \ttie0\qb0{o}\itieu0o\tqb0p\ttie0\en \endextract % 結束記譜 \end{music}</pre>
迷宮	labyrinth		<pre>\begin{labyrinth}[unit=7.5pt]{19}{20} % 繪製迷宮 \h *9+---*7+ \v +*5-+-+-----+ \h -*4+-----*4+--+ \v +*4-++-+-----+ \h *4+-----+----+ \v +-----+-----+ \h -+-+-----*4-+++ \v +-+-+-----+ \h ++++-----+ \v +-----*4--*4+--+ \h -+++*4+--+*6-+ \v +-----+-----+ \h ++++-----+ \v +-----+-----+ \h -++++-----+ \v +-+-----+-----+ \h +-----+-----+ \v +-----+-----+ \h -*7-++*4-+++ % (略...) % 繪製解法路徑 \autosolution[font=\color{red}](8,0)(10,19){u} \end{labyrinth}</pre>
填字遊戲	crossword (cwpuzzle)	<p>橫向</p> <p>1 一種 2019 年開始大流行的疾病</p> <p>2 一個排版軟體</p> <p>3 一套排版引擎</p> <p>縱向</p> <p>2 作者讀的高中</p>	<pre>\begin{Puzzle}{5}{5} [1]C O V I D . [2]W O R D { }. [3]L A T E X . S { } { } { }. H { } { } { }. \end{Puzzle}</pre>

▲ 表 1.2: L^AT_EX 的趣味應用範例

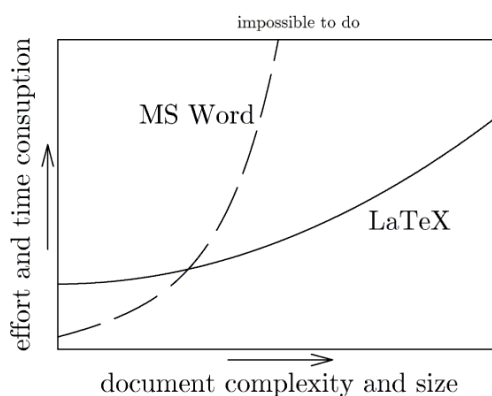
1.5 L^AT_EX 與其他軟體的比較

看了 L^AT_EX 這麼精采的表現，此時的你可能有點心動，想著手學習這套炫炮的程式了。但看到密密麻麻的程式碼與指令，你又猶豫了一下，覺得自己從小到大用 Word 用得好好的，現在幹嘛花一堆時間折磨自己。為了更清楚 L^AT_EX 與 Word 的差別，讓我們看看下表：

軟體 屬性	L ^A T _E X LaTeX	 Microsoft Word
價錢	免費	付費
難易度	難	易
專業度	高	低
多人合作	難	易
處理圖片	操作複雜，位置穩定	操作簡單，位置易亂跑
編輯模式	需將原始碼編譯成文件	WYSIWYG ¹
適合用於	大型、複雜的文件	小型、簡單的文件

▲ 表 1.3: L^AT_EX vs. Microsoft Word

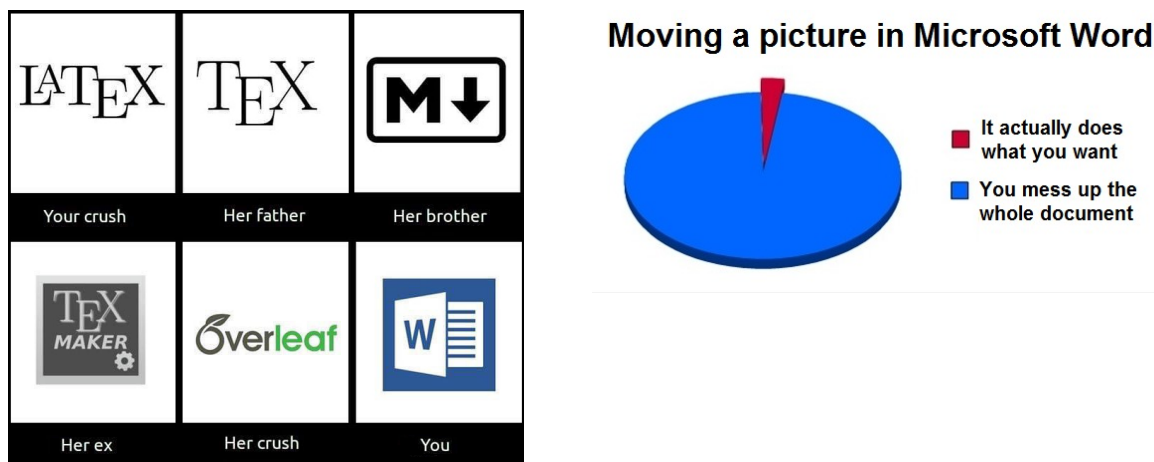
接著我們來看一張曲線圖。由圖中的兩條曲線可明顯得知，當我們在處理簡單的文件時，用 Word 確實比 L^AT_EX 省時；但隨著文件的大小和複雜度提升，使用 L^AT_EX 反而比 Word 來得輕鬆。



▲ 圖 1.3: 利用 L^AT_EX 與 Word 處理不同複雜度的文件時，分別所需的時間與精力
















¹What You See Is What You Get, 所見即所得

也正因為兩套軟體的性質迥異，許多梗圖就因此誕生...



▲ 圖 1.4: 更多關於 L^AT_EX 與 Word 的梗圖

最後，既然說 L^AT_EX 與程式語言的運作邏輯比較相似，那就來與時下當紅的 C++ 與 Python 比較吧！

語言 使用軟體/屬性	L ^A T _E X LaTeX	C++	Python
編輯器 (Editor) / 整合開發環境 (IDE ²)	<ul style="list-style-type: none"> • Texmaker  • TeXstudio  • TeXworks  	<ul style="list-style-type: none"> • Dev-C++  • Code::Blocks  	<ul style="list-style-type: none"> • PyCharm  • Spyder  • Jupyter 
編譯器 (Compiler) / 直譯器 (Interpreter)	<ul style="list-style-type: none"> • pdfL^AT_EX • X_LL^AT_EX • LuaL^AT_EX 	<ul style="list-style-type: none"> • G++  • MSVC³  	<ul style="list-style-type: none"> • CPython  • PyPy 
輸出 (Output)	.pdf 	.exe 	(.exe )

▲ 表 1.4: L^AT_EX vs C++ vs Python

²Integrated Development Environment, 整合開發環境

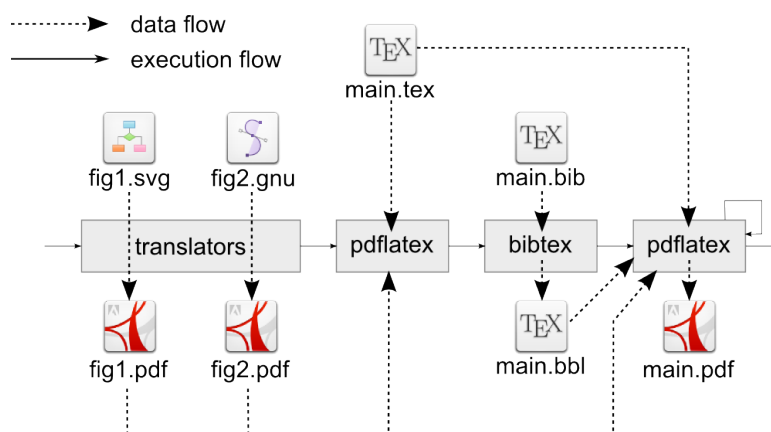
³Microsoft Visual C++

第 2 章

初始環境設定

2.1 L^AT_EX 的運作模式與編譯流程

先前提到 L^AT_EX 是一套功能強大且複雜的排版「系統」，顧名思義可知它並非一個獨立運行的程式，而是將每個獨立的小程式串成一組「編譯鍊」(Compile Chain)，才得以支援這麼多自動化的功能。



▲ 圖 2.1: 一種典型的 L^AT_EX 文件編譯流程

一份文件的產出還必須仰賴許多字形檔、格式檔與額外的套件包，而套件包之間的相依性也往往非常複雜，不同的文件還可能會用到不同的編譯流程。因此，這些工具不可能讓我們一個一個安裝，因為保證還沒裝完你就會先累死。

幸運的是，科技就是為了便利而存在。網路上早已有很多大佬設計的 T_EX/L^AT_EX 發行版 (Disturbution)，裡面就包含了那些最常用的工具，免去手動安裝的煩惱。而且它還有另一個最重要的功能，也就是會自動管理需要用到的套件，缺了什麼就幫你下載什麼，根本就是最偉大的發明之一了 (XD)。(真不愧是號稱系統管理者的救世主)

※ 其實許多大型程式的開發也同樣必須透過一連串的工具達成，例如一套典型的開發環境可能會包含：文字編輯器、編譯器、連結器、函式庫、偵錯工具、效能分析工具等等。

※ 只要是多少聽過或用過 Linux 的人，也一定對發行版與套件包管理工具的概念不陌生。Python 中的 pip 或 conda 也是一種套件包管理工具的範例。

2.2 安裝 L^AT_EX 發行版 — MiK_TE_X

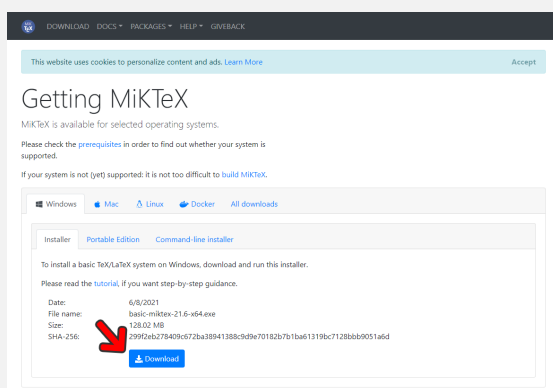
貓貓在本書中皆使用 Windows 作業系統作為示範，而支援 Windows 的發行版主要有 MiK_TE_X 與 T_EX Live。其實因為各發行版的功能差異很小，再加上貓貓懶得比較兩者的差異，所以就選擇最多人用的 MiK_TE_X 作為示範吧...

步驟

圖示

說明

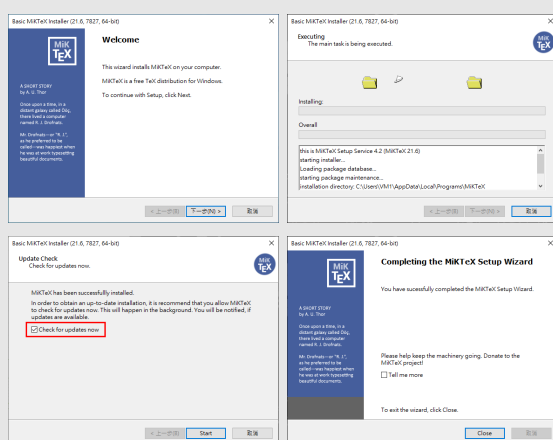
1.



至 [MiK_TE_X 官方網站](#) 下載最新的 MiK_TE_X 發行版。

以 Windows 系統為例，檔名應為 basic-miktex-(版本)-x64.exe。

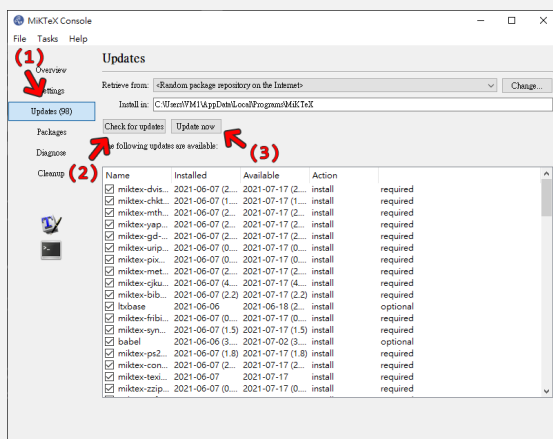
2.



執行安裝程式，之後就是熟悉的連點下一步 (XD)，並等待安裝程式複製完所有檔案。(預設的設定應該能符合大多數人的需求啦)

複製完檔案後，記得將 ☒ Check for updates now 打勾，讓套件管理工具檢查是否有套件更新。接著就可以一直下一步，直到安裝程式結束。

3.



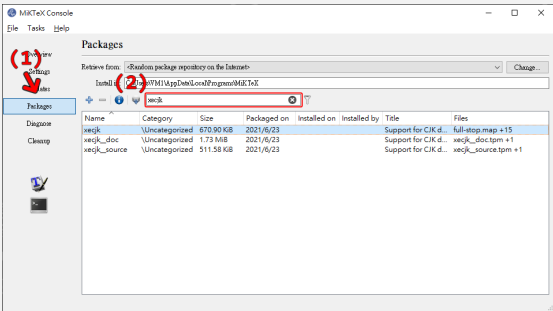
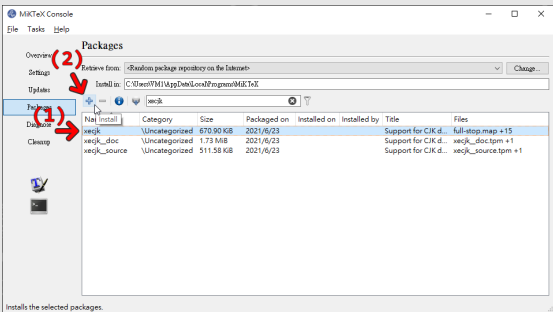
(1) 開啟 MiK_TE_X Console，並點擊左側的 Updates。

(2) 若下面的列表為空的 (或是為了保險起見)，點擊 Check for updates。

(3) 等待程式檢查更新後，點擊 Update now，並更新所有可更新的套件。

2.3 安裝套件 — xeCJK

安裝完發行版後，每個 L^AT_EX 人都一定要會的就是安裝套件啦！為了讓 L^AT_EX 可以支援中文輸入，本書就以 xeCJK 套件作為範例。這應該也是大家必須且最常用的套件之一，所以現在就馬上裝起來吧！

步驟	圖示	說明
1.		<p>(1) 開啟 MiKTeX Console，並點擊左側的 Updates。</p> <p>(2) 在畫面中間的搜尋欄內輸入“xecjk”後，按下 。</p>
2.		<p>(1) 點選視窗中第一列的“xecjk”。</p> <p>(2) 點擊搜尋欄左側的  圖示 (Install)，進行套件安裝。</p>
3.		<p>等待套件安裝完畢後，點擊視窗右下角的 Close。</p>

※ 偷偷跟你說，其實不一定每次安裝套件都要這麼麻煩。假如你像我一樣很懶，日後直接在文件裡面用了 `\usepackage{還沒安裝的套件}`，你的編譯器在編譯時應該會直接問你要不要安裝該套件，到時候直接按 Install 就解決了，呵呵。

2.4 安裝 L^AT_EX 編輯器 — Texmaker

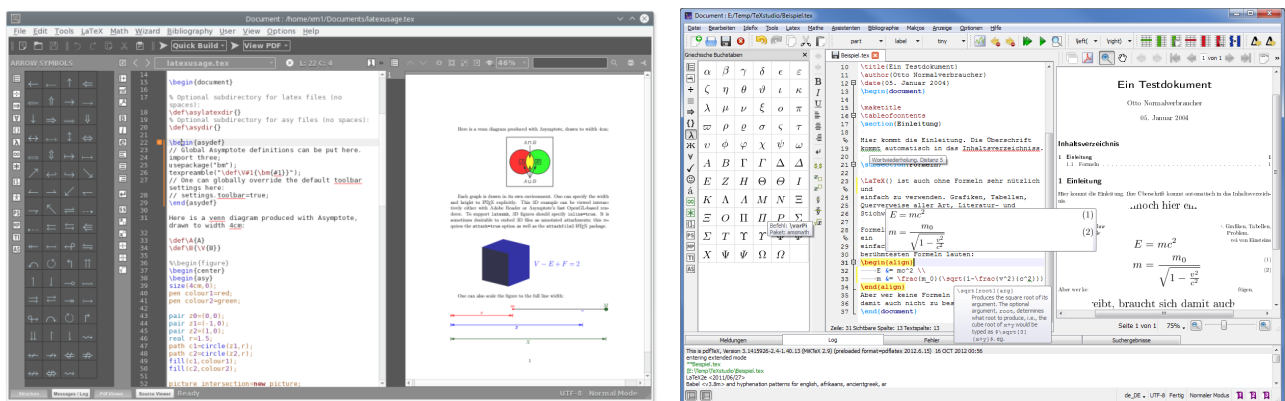
雖然先前提到 L^AT_EX 的相容性極高，在任何能打出字的地方都可以直接編輯 (包括 Windows 內建最陽春的記事本)，但為了我們的方便，選擇一個專為 L^AT_EX 設計的文字編輯器還是可以省下許多麻煩，例如自動完成指令、快速建表、快速插入圖片、格式設定快速鍵，還有最重要的是不用每次都要手動到終端機下編譯指令。

L^AT_EX 的文字編輯器有分為本機程式和線上網站兩種。首先，貓貓強烈不推薦線上編輯器。雖然它們可能會提供多人同時編輯的服務，但大多必須犧牲套件的選擇多元性，而且很多「進階」的功能還要付費，根本就划不來。Overleaf 就是最有名的線上編輯器之一。雖然它的編輯器不好用，但我覺得它的教學文很值得參考。貓貓當年也是在讀書會學長的推薦下，看它的 [Learn L^AT_EX in 30 minutes](#) 學會 L^AT_EX 的基礎。

就本機程式的選擇而言，其實我們之前安裝的 MiK_TE_X 就有附加一個名為 TeXworks 的編輯器了，但大部分的人都覺得它功能太陽春、不太實用，因此目前網上主流的選擇包括 Texmaker、TeXstudio、WinEdt 等等。其實它們的功能都差不多，可是貓貓感覺大家比較喜歡 Texmaker，大部分的教學文也都是用它作為範例。我平常都是用 TeXstudio 啦，純粹只是因為我覺得它在我的電腦上明顯比較好看、設定比較合我胃口而已。

順道一提，Texmaker 有支援繁體中文，但 TeXstudio 目前只有支援簡體中文，所以很討厭簡體字的我把所有東西都調成英文。但考慮大家可能無法接受全英文的介面，本書還是選擇用 Texmaker 作為示範，畢竟最多人用的東西應該不會出什麼大問題才對 (希望啦)。

安裝文字編輯器的流程與其他軟體大同小異，相信你們都可以的，我就不在此贅述了。我們就直接進到重點的下一章，也就是來設定文字編輯器。



▲ 圖 2.2: L^AT_EX 編輯器的截圖 (左)Texmaker (右)TeXstudio

2.5 設定 Texmaker

到此為止，大家都把要用的東西都安裝好了，最後只剩下一點點設定的步驟就完成了！

接下來我們必須決定要用哪個編譯引擎，而目前主流的 \LaTeX 編譯器包括 pdf\LaTeX 、 Xe\LaTeX 、 Lua\LaTeX 等等。貓貓看到華語圈的大家幾乎都是用 Xe\LaTeX 作為編譯引擎，可是因為比起較舊的那些編譯引擎， Xe\LaTeX 對 Unicode 與 OpenType 等多國語言和字型功能的支援度較廣泛，也就意味著製作英文以外的文件就必須靠它了。還記得先前安裝的 xeCJK 嗎？它也是為了配合 Xe\LaTeX 讀取中文輸入所設計出的套件。

因此，我們必須叫 Texmaker 不要傻傻呼叫預設的 \LaTeX 編譯引擎，而是用我們希望的 Xe\LaTeX 。

步驟	圖示	說明
1.		<p>(1) 開啟 Texmaker，點選上方工具列的“選項(O)”。</p> <p>(2) 點擊第一個“設定 Texmaker”，叫出設定視窗。</p>
2.		<p>(1) 點選左側的“快速編譯”，進入快速編譯的視窗。</p> <p>(2) 選擇“XeLaTeX + View PDF”為快速編譯命令。</p>

恭喜大家，我們漫長的前言終於到此告一段落啦！在本書的下一部份中，貓貓就要帶大家來實際上機操作，寫出你的第一份 \LaTeX 文件！

Part II

L^AT_EX 入門篇

L^AT_EX for Beginners

第 1 章

寫出你的第一份 L^AT_EX 文件！

1.1 Hello, World! in L^AT_EX

```
1 \documentclass{article}
2 \begin{document}
3     Hello, World!
4 \end{document}
5
```

程式碼 1.1: Hello World! in L^AT_EX

Part III

L^AT_EX 進階篇

L^AT_EX for Intermediates

Part IV

L^AT_EX 專精篇 L^AT_EX for Experts