

# 第 1 章

## L<sup>A</sup>T<sub>E</sub>X 簡介

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

### 1.1 什麼是 T<sub>E</sub>X？

T<sub>E</sub>X 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<sub>E</sub>X 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<sub>E</sub>X 後續衍生出的一大堆程式區別，這種最早出現、最陽春的 T<sub>E</sub>X 也經常被稱為原版 T<sub>E</sub>X (Plain T<sub>E</sub>X)。

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

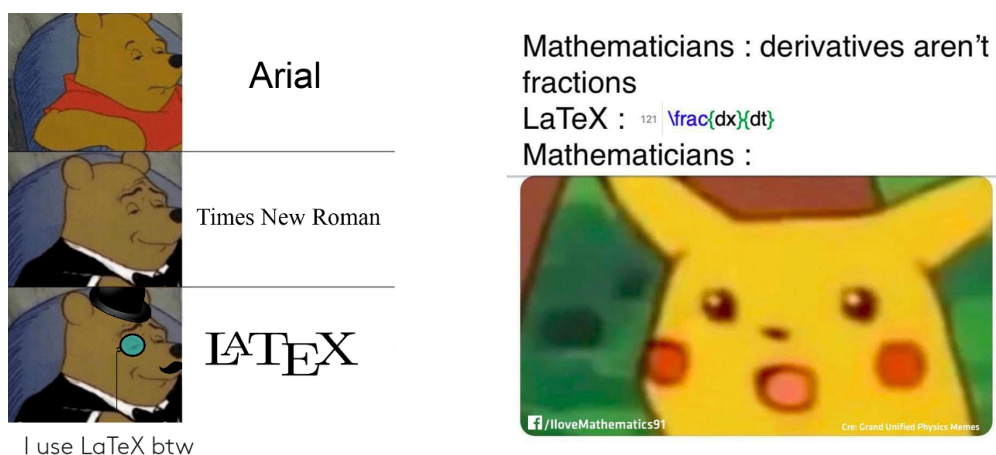
隨著科技的發展，當年的原版 T<sub>E</sub>X 所提供的功能早已不敷使用，同時也被人覺得太複雜、不親民。因此，美國又有一位電腦科學家 Leslie Lamport 在 1984 年發表了基於 T<sub>E</sub>X 的排版系統 L<sup>A</sup>T<sub>E</sub>X，支援更多實用的功能和更親民的指令集，也推廣了這整套系統的應用。

講白話一點，L<sup>A</sup>T<sub>E</sub>X 其實就是 T<sub>E</sub>X 的 PRO 版，而且比原本的 T<sub>E</sub>X 好用 n 百倍，導致 T<sub>E</sub>X 已經被大眾打入冷宮了。

### 1.3 為什麼厲害的人都在用 L<sup>A</sup>T<sub>E</sub>X ?

L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 的特色與優點：

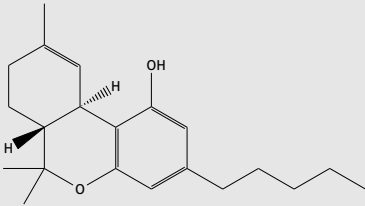
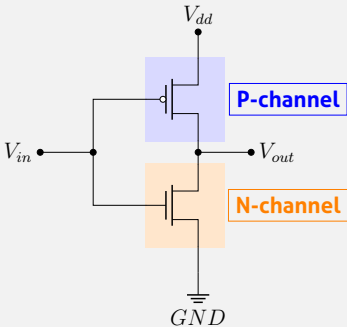
- **數學公式間距調整**：L<sup>A</sup>T<sub>E</sub>X 幾乎就是為了排版數學公式而生的，其內部的演算法可配合當前字型，自動調整數學公式中數字與符號間的距離；使用者也可用指令手動增加或減少間距大小。
- **萬用的純文字檔案**：L<sup>A</sup>T<sub>E</sub>X 的原始碼以純文字檔案儲存，因此檢視與編輯時不受作業系統或裝置限制，只要是打得出字的機器都能直接編輯。假設你的檔案是純英文，甚至可以用摩斯電碼傳給你朋友。
- **一勞永逸的格式設定**：一份 L<sup>A</sup>T<sub>E</sub>X 文件的所有格式設定都是透過指令達成的，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<sub>E</sub>X 大家族的軟體都是完全免費的，當然也包括 L<sup>A</sup>T<sub>E</sub>X 與大部分的實用套件 (Package)。
- **廣大的套件支援**：L<sup>A</sup>T<sub>E</sub>X 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 1.1: 兩張關於 L<sup>A</sup>T<sub>E</sub>X 的梗圖

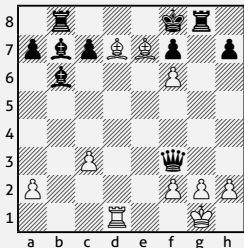
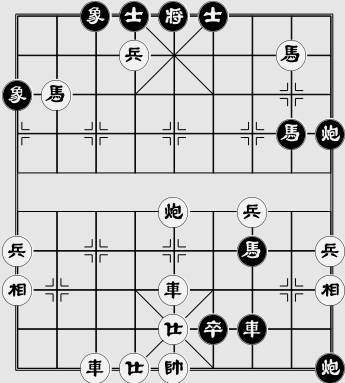
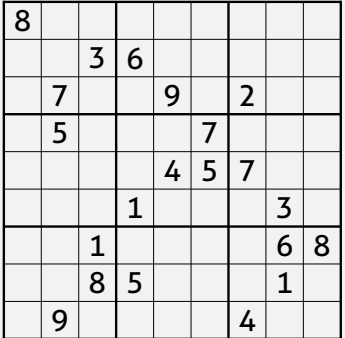
# 1.4 L<sup>A</sup>T<sub>E</sub>X 究竟有多強大？

說了這麼多，就讓我們實際看看 L<sup>A</sup>T<sub>E</sub>X 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> \[\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx\] ↪ 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(-&lt;[:1]H) (*6(-(-)-)))-(&lt;[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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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	
-----	----------	--

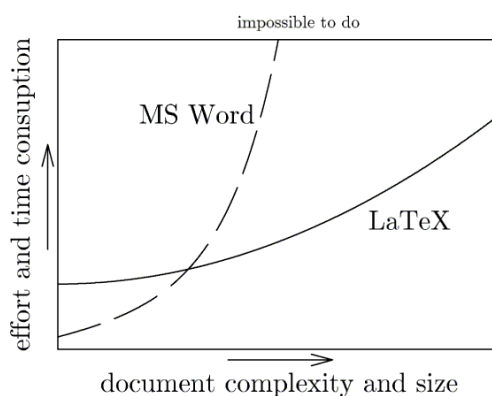
## 1.5 L<sup>A</sup>T<sub>E</sub>X 與其他軟體的比較

看了 L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 與 Word 的差別，讓我們看看下表：

軟體 屬性	L <sup>A</sup> T <sub>E</sub> 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 <sup>1</sup>
適合用於	大型、複雜的文件	小型、簡單的文件

▲ 表 1.3: L<sup>A</sup>T<sub>E</sub>X vs. Microsoft Word

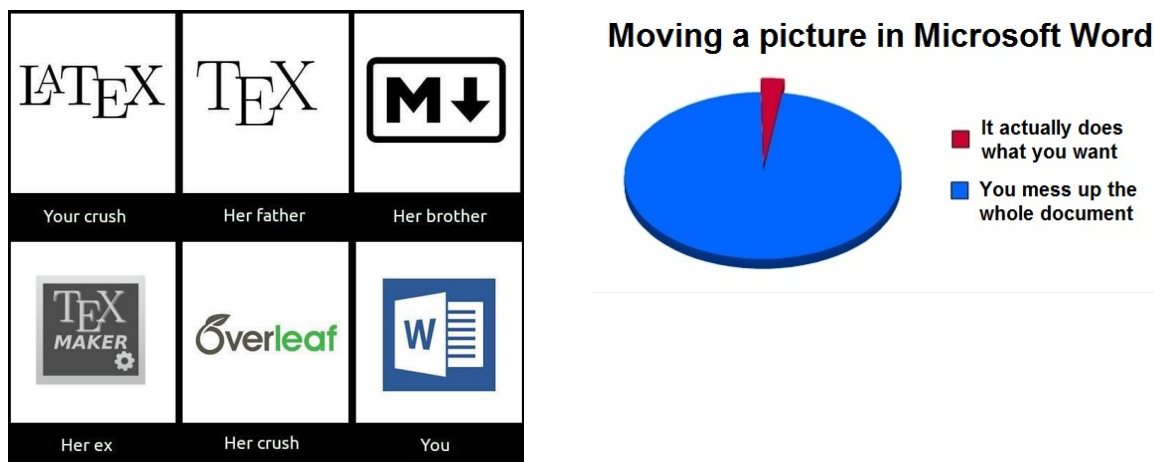
接著我們來看一張曲線圖。由圖中的兩條曲線可明顯得知，當我們在處理簡單的文件時，用 Word 確實比 L<sup>A</sup>T<sub>E</sub>X 省時；但隨著文件的大小和複雜度提升，使用 L<sup>A</sup>T<sub>E</sub>X 反而比 Word 來得輕鬆。



▲ 圖 1.2: 利用 L<sup>A</sup>T<sub>E</sub>X 與 Word 處理不同複雜度的文件時，分別所需的時間與精力
















<sup>1</sup>What 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 1.3: 更多關於 L<sup>A</sup>T<sub>E</sub>X 與 Word 的梗圖

最後，既然說 L<sup>A</sup>T<sub>E</sub>X 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 <sup>A</sup> T <sub>E</sub> X LaTeX	C++	Python
編輯器 (Editor) / 整合開發環境 (IDE <sup>2</sup> )	<ul style="list-style-type: none"> <li>• Texmaker </li> <li>• TeXstudio </li> <li>• TeXworks </li> </ul>	<ul style="list-style-type: none"> <li>• Dev-C++ </li> <li>• Code::Blocks </li> </ul>	<ul style="list-style-type: none"> <li>• PyCharm </li> <li>• Spyder </li> <li>• Jupyter </li> </ul>
編譯器 (Compiler) / 直譯器 (Interpreter)	<ul style="list-style-type: none"> <li>• pdfL<sup>A</sup>T<sub>E</sub>X</li> <li>• X<sub>L</sub>L<sup>A</sup>T<sub>E</sub>X</li> <li>• LuaL<sup>A</sup>T<sub>E</sub>X</li> </ul>	<ul style="list-style-type: none"> <li>• G++ </li> <li>• MSVC<sup>3</sup> </li> </ul>	<ul style="list-style-type: none"> <li>• CPython </li> <li>• PyPy </li> </ul>
輸出 (Output)	.pdf 	.exe 	(.exe  )

▲ 表 1.4: L<sup>A</sup>T<sub>E</sub>X vs C++ vs Python

<sup>2</sup>Integrated Development Environment, 整合開發環境

<sup>3</sup>Microsoft Visual C++