

“TeX” Support Proposal

Lorenzo Hess
2024-12-19

Contents

1	Introduction	3
2	What’s Our Agenda?	3
3	What is L^AT_EX?	3
4	Feature Comparison	4
4.1	The General WP	4
4.2	L ^A T _E X	4
4.2.1	Don’t Switch to L ^A T _E X If...	4
4.3	Table of Features	4
4.3.1	L ^A T _E X vs. General WP	5
4.3.2	Overleaf vs. Google Drive vs. Microsoft Onedrive	5
5	How Tos	5
5.1	How to Read Error Messages	5
5.2	How to Debug L ^A T _E X Code	5
5.3	How to Use L ^A T _E X Package Documentation	5
5.4	How to Find Help	5
5.5	How to Use Detexify	5
6	Programming Tips	5
6.1	Naming Conventions	5
6.1.1	Sections	5
6.1.2	Labels	5
6.2	How To Manually Modifying Spacing	5
6.3	L ^A T _E X Editing Workflow	5

7	L^AT_EX Editors	6
8	Overleaf for Collaborative Editing	6
9	Inkscape and L^AT_EX For Fancy Diagrams	6
10	Custom Code Snippets	6

1 Introduction

This document outlines ideas for a comprehensive approach to L^AT_EX advocacy, covering everything from pro-L^AT_EX talking points, arguments against word processors (WPs), solutions for group work in L^AT_EX, and L^AT_EX code snippets, including full preambles and examples. While any individual could make use of this advocacy material, a group or club would be ideal in terms of providing additional resources and labor. Such a group could be called “TeX Support”, so I’ve used that in the title of this document ¹.

2 What’s Our Agenda?

We like L^AT_EX, and we think it’s better for preparing documents, in most cases, than Word Processors (WPs). We’re also pragmatic, and recognize that many people haven’t used L^AT_EX or may have felt frustrated when they have. Many L^AT_EX tutorials indeed fail to enumerate and provide what we see as essential foundations for learning, appreciating, and enjoying L^AT_EX document preparation. For example, many tutorials characterize L^AT_EX’s utility in terms of what it is good at, such as typesetting math or providing robust label and reference management. They fail, however, to provide a comprehensive feature comparison between L^AT_EX and WPs that would make clear L^AT_EX’s superiority. Tutorials also generally fail to fully compare the L^AT_EX and WP mindsets, and thus fail to help users realize how WPs hinder writing. Finally, tutorials fail to provide essential tools that make L^AT_EX editing less tedious, from code templates like preambles to software like Detexify.

We have two general goals: 1) present a comparative, serious, and evidence-based argument for when, and why, individuals and groups should use L^AT_EX instead of WPs; and 2) to provide code templates, code snippets, and guides for individuals and groups to not only get started with L^AT_EX, but use it to its full potential.

We’re not here to evangelize. WPs do have certain advantages over L^AT_EX (see the feature comparison in Section X), and it’s a free country (and even freer Internet): we don’t care what software you prefer or use. We do believe that L^AT_EX, combined with a proper editor and editing workflow, usually makes writing and formatting faster and better.

3 What is L^AT_EX?

What is LaTeX? What is a markup language? Do I need to know programming? Is LaTeX programming?

You write instructions (i.e. code) to tell a latex compiler how to prepare your PDF, i.e. how to format, structure, and generally manipulate text, images, tables, etc to make your PDF.

¹Thanks to Keegan Kuhn for this pun.

Analogies: latex is the movable type of PDFs.

4 Feature Comparison

4.1 The General WP

First cover features, operation (GUI: menus, buttons, and keybinds), and general workflow. Cover the WP mindset.

4.2 L^AT_EX

First cover features, operation (TUI: writing code and compiling), and general workflow. Cover the latex mindset.

4.2.1 Don't Switch to L^AT_EX If...

Don't switch to LaTeX if

1. You're not willing to learn the minimum necessary for your needs (might just be math equations, or might extend to bibliographies).
2. All this being said, LaTeX is just text (another benefit!). If you've started a project but can't finish it in LaTeX, you can always copy/paste it somewhere else, or even export it directly to docx with tools like Pandoc (won't export everything perfectly).

4.3 Table of Features

We should quantify as many features as possible and present them in a table for a comprehensive feature comparison.

4.3.1 \LaTeX vs. General WP

4.3.2 Overleaf vs. Google Drive vs. Microsoft Onedrive

5 How Tos

5.1 How to Read Error Messages

5.2 How to Debug \LaTeX Code

5.3 How to Use \LaTeX Package Documentation

5.4 How to Find Help

tex stackexchanger, overleaf.com/learn, google searches

5.5 How to Use Detexify

Really simple.

6 Programming Tips

6.1 Naming Conventions

6.1.1 Sections

Good practices for naming sections

6.1.2 Labels

Good practices for naming labels, e.g. `sec.subsec.subsubsec` (dot notation) or `sec:subsec::subsubsec` (colon notation).

6.2 How To Manually Modifying Spacing

How to use positive and negative `vspace` and `hspace`. How to use `vfill` and `hfill`.

Macros for inserting a preset amount of horizontal space, e.g. to separate answers within a box

6.3 \LaTeX Editing Workflow

Your own preamble. File templates. Homework templates. Bibliographies

7 L^AT_EX Editors

TUI (emacs and vim), VSCode, overleaf. Find other popular ones to cover.

8 Overleaf for Collaborative Editing

Don't give Overleaf tutorial (they have docs for that). We should outline how to manage group projects and collaborative writing in latex.

9 Inkscape and L^AT_EX For Fancy Diagrams

10 Custom Code Snippets

1. Header and footer for nice homework.
2. Two figures side by side.
3. Appendices
4. Bibliography
5. acronyms
6. landscape pages
7. block quotes
8. list spacing
9. actually put floats where you want them
10. different fonts and math fonts
11. footnotes
12. glossaries
13. page margins
14. date formatting
15. multi-row table cells
16. code styling
17. include PDF pages
18. charts and other graphs
19. justification
20. line spacing
21. reduce vertical space between word and underline
22. vertical table cells
23. URL colors
24. single space table of contents