Docs - LaTeX helpers

Janis Hutz
https://janishutz.com

October 19, 2025

Contents

1 Introduction

This set of LaTeXfiles is designed to give you a good looking, pre-configured LaTeXsetup, which helps you get started much more quickly.

It has some configuration options already, but more are to come soon. If you have any suggestions as to what should be added, don't hesitate to open a support ticket at `https://support.janishutz.com?a=add` or contacting me via email to `development@janishutz.com`.

2 Breaking Changes

Please note that the regex provided below are for NeoVim and you may need to replace `\(\)` with `()`

2.1 Time complexity

The timecomplexity commands have been updated to require math environment. You can use the following regex to fix them

`/ \\tc\([olt]\){\(.*)}\([, .\n]\)/ \$\\tc\1{\2}\$\3/g` (execute first)
and `/\\text{\\tc\([olt]\){\(.*\)}}/\\tc\1{\2}/g`

2.2 Labels & Environments

The counter environments can now be referenced (see 6.6.3). Due to the new requirements set out by that addition, it is now no longer possible to manually update the counters using `\setcounter`. Use the below regex to replace all occurrences of `\setcounter` with the new format. Please note that this will overwrite *all* occurrences of `\setcounter` with the specific old format with `\setLabelNumber`:

`/\\setcounter{\(.*\)s}/\\setLabelNumber{\1}/g` and `/\\setcounter{all}/\\setLabelNumber{all}/g`

and for the `\stepcounter` as well

`/\\stepcounter{\(.*\)s}/\\stepLabelNumber{\1}` and `/\\stepcounter{all}/\\stepLabelNumber{all}/g`.

3 Installation

You can install these helper files by downloading this repo and storing it to any location on your PC, remembering where that location is.

You may also install the VSCode snippets found in the vscode-snippets folder. These snippets provide autocompletion for many of the commands that this helper file provides.

4 Usage

You can type `latex-prepare` and press tab, if you have installed the VSCode snippets, or copy over this code snippet:

```
\documentclass{article}

\newcommand{\dir}{~/path/to/helper} % TODO: Change your path here! No trailing slashes!
\input{\dir/include.tex}
\load{recommended} % TODO: Change the inclusion level (if necessary), see below

\setup{Type your title here}

\begin{document}
\startDocument

Type your \LaTeX here

\end{document}
```

The `\load` command takes one parameter, which can be one of the following:

- `minimal` Just the core styling and core functionality.
- `most` Includes Math, CS, Language and all styling.
- `recommended` The recommended setup. Includes BibTeX in addition to what is in most
- `all` If you want to also include glossaries or code with highlighting
- `letter` If you want to typeset a letter (see 4.2)

4.1 Configuration

You can set a global config in config file in helper files directory. The file is located at

`/<path to helpers>/config/config.tex`. All configs are documented there. Simply open that file using any text editor and edit your config.

4.2 Letters

Letters require a different setup compared to a normal LaTeXdocument:

```
\documentclass[12pt,a4paper]{scrlttr2}

\newcommand{\dir}{~/path/to/helper} % TODO: Change your path here! No trailing slashes!
\input{\dir/include.tex}
\load{letter}

\setkomavar{subject}{} % type your subject here
\begin{document}\raggedright
    % below, type the address, being careful not to remove the backslashes
    \begin{letter}{Company \\ Name \\ Address \\ CH-Place}
        \opening{Intro}

        Your text

        \closing{Kind regards} % Your closing sentence
    \end{letter}
\end{document}
```

If you are using the snippets, you can type `latex-letter` and press tab.

5 Custom Loader

You can create a custom loader by defining a `load{`*string*`}` macro (or any other, if you are aware you need to change that in your main file for it to work) that uses the `\dir` command to resolve the custom parts. Take a look at the `/include.tex` file for inspiration on how to do it, if you also want it to do case distinction.

You can also include some the pre-built configs from the `dist/` folder.

6 Full Command Reference

6.1 Variables

- *scope*: `minimal`, `most`, `recommended`, `all` or `letter`
- *string*: Any normal text
- *math*: Any math input
- *number*: Any non-negative integer, i.e. no commas
- *color*: Any of the `dvipsnames` colours of `xcolor`
- *langauge*: 2-character country code (currently only `de` and `en` supported)

6.2 Per-File config

- `\renewcommand{\authorTitle}`{*string*} Change the author (in the title) for this document only
- `\renewcommand{\authorHeaders}`{*string*} Change the author (in the header) for this document only
- `\renewcommand{\name}`{*string*} Change the name (for letters) for this document only
- `\renewcommand{\street}`{*string*} Change the street (for letters) for this document only
- `\renewcommand{\city}`{*string*} Change the city (for letters) for this document only
- `\renewcommand{\countrycode}`{*string*} Change the country-code (for letters) for this document only
- `\renewcommand{\theoremde}`{*string*} Change the translation for theorem in German (usually either "Satz" or "Theorem")
- `\setNumberingStyle`{*number*} Change the numbering of definitions, lemmas, etc for this document. 0 = off, 1 = Separately, 2 = Combined (except for definition), 3 = Combined
- `\setcounter{numberSubsections}`{*number*} Change the format of the numbering of definition, lemma, etc.
  0 = <section>.<number>,
  1 = <section>.<subsection>.<number>,
  2 = <section>.<subsection>.<subsubsection>.<number>
  Changing this won't affect all of numbering prior to the command, only after. You can change this setting (and the one above) at any point in the document
- `\setcounter{descriptorShadeStrength}`{*number*} Change the colour saturation of the inline descriptors
- `\setcounter{shadeStrength}`{*number*} Change the colour saturation of the `\shade` command
- `\setLang`{*language*} Change the language. Will automatically load babel in German. Can only be used once and only at the start of the document or in the preamble, as repeated usage leads to undefined behaviour
- To change the font for the entire document, load the font package using `\usepackage`, with the last occurrence of a font package determining the active one. Then select the type by using `\setFontType`{*mono* | *serif* | *sans*}. A list of fonts is available here. You may change the font only for a specific section, by enclosing `\setFont` and the text that should be written in said typeface in curly braces.

6.3 Setup, Loading & Translation

- `\load`{*scope*} Load the selected *scope*
- `\setup`{*string*} Prepare the document with the *string* being the title
- `\setupCheatSheet`{*string*} Prepare the document with smaller borders and no headers / footers. *string* is the title
- `\setupBarebones`{*string*} Minimal setup, only borders and title set
- `\startDocument` Initialize the document. Has to be called after `\begin{document}`
- `\usetcolorboxes` Initialize tcolorboxes. In main body, if you want to use fancy boxes. (requires `most` or up)
- `\translate`{*string*}{*string*} First *string* is English, second *string* is German. Switches automatically based on language selected
- `\tr`{*string*}{*string*} Shorthand for `\translate`
- `\numberingOn` Turn on the numbering (will set back to config previously set by `\setNumberingStyle`)
- `\numberingOff` Turn off the numbering (if you want to temporarily not use it. Do not use `\setNumberingStyle` for that)

6.4 Math-Commands

All these have to be executed in the math environment.

- `\R` Prints $\mathbb{R}$. Same goes for `\C` printing $\mathbb{C}$, etc.
- `\floor{`*math*`}` Round down symbol, e.g. $\lfloor n \rfloor$
- `\ceil{`*math*`}` Round up symbol, e.g. $\lceil n \rceil$
- `\hastoeq` or `\mbeq` Has to equal symbol (non-standard), $\overset{!}{=}$
- `\Leftrightarrowequiv` Equivalence transformation symbol, $\dot{\Leftrightarrow}$.
- `\Rightarrowequiv` Equivalence transformation symbol, $\dot{\Rightarrow}$.
- `\Leftarrowequiv` Equivalence transformation symbol, $\dot{\Leftarrow}$.
- `\defAs` Define as, i.e. $\overset{\text{def}}{=}$
- `\defEquiv` Define as, but with a two-sided implication instead of equality, i.e. $\overset{\text{def}}{\Leftrightarrow}$
- `\defImplies` Define as, but with one-sided implication, i.e. $\overset{\text{def}}{\Rightarrow}$
- `\divides` Divider or divides symbol, e.g. $a \mid b$
- `\lcm` Least common multiple, lcm
- `\arcsinh` Inverse of hyperbolic sine, arcsinh
- `\arccosh` Inverse of hyperbolic cosine, arccosh
- `\arctanh` Inverse of hyperbolic tangent, arctanh
- `\limit{`*math*`}{`*math*`}` Shortened limit notation, $\lim\limits_{x \to x_0}$
- `\limni` Shortened limit notation for $n \to \infty$, $\lim\limits_{n \to \infty}$
- `\liminfni` Shortened limit inferior notation for $n \to \infty$, $\liminf\limits_{n \to \infty}$
- `\limsupni` Shortened limit superior notation for $n \to \infty$, $\limsup\limits_{n \to \infty}$
- `\der{`*math*`}` Derivative, $\dfrac{\mathrm{d}}{\mathrm{d}x}$
- `\dern{`*math*`}{`*math*`}` Higher derivative, $\dfrac{\mathrm{d}^2}{\mathrm{d}^2 x}$
- `\elementstack{`*math*`}{`*math*`}` Stack two elements on top of eachother. Uses `\genfrac` under the hood. Can be used for example in limits as an alternative to `\atop` or `\substack`.

6.5 CS-Commands

These commands have to be executed inside math environment, except `\timecomplexity` and `\tc`.

- `\timecomplexity` Prints the word time complexity with a coloured box.
- `\tc` Shorthand for `\tct`. Deprecated
- `\tct` Time complexity in $\Theta()$-notation (average case)
- `\tco` Time complexity in $\mathcal{O}()$-notation (worst case / upper bound)
- `\tcl` Time complexity in $\Omega()$-notation (best case / lower bound)
- `\t[R, C, N]` Simply prints a capital R, C or N in math mode as normal text. So e.g. use R to print an R
- `\wordbool` Prints $(\Sigma_{\text{bool}})^*$
- `\words{`*string*`}` Prints $(\Sigma_{\text{test}})^*$
- `\wordm{`*math*`}` Prints Σ_1^*
- `\word` Prints Σ^*
- `\alphabets{`*string*`}` Prints Σ_{test}
- `\alphabetbool` Prints Σ_{bool}

Algorithms

```
\begin{algo}{functionName(A)}
    \Procedure{functionName}{$(A)$}
        \State\Return "Hello World"
    \EndProcedure
\end{algo}
```

Algorithm 1 FUNCTIONNAME(A)

```
1 procedure FUNCTIONNAME((A))
2     return "Hello World"
```

6.6 Style

General styling commands. All other commands, except these ones require at least `most` to be the selected *scope*

- `\TODO` Print a highlighted TODO:
- `\background`{*color*}{*number*}{*string*} Print **shaded text, with colour saturation**
- `\shade`{*color*}{*string*} Print **shaded text**
- `\backdrop`{*string*} Print **text with gray backdrop**
- `\fhlc`{*color*}{*string*} Print **bold, underlined text in a coloured box**
- `\fhl`{*string*} Print **bold, underlined text in a white box**
- `\printtoc`{*color*} Print the table of contents (as seen on the first page). The normal `\tableofcontents` still works as expected
- `\smallhspace` Prints a 2mm hspace
- `\mediumhspace` Prints a 5mm hspace
- `\largehspace` Prints a 10mm = 1cm hspace
- `\rmvspace` Removes some of the vertical whitespace printed by environments like `align`
- `\drmvspace` Removes some of the vertical whitespace printed by environments like `align` (double the amount)

6.6.1 Tcolorboxes

Included in `most` and up

Put `\usetcolorboxes` right after `\startDocument` (right after `\begin{document}`) if you plan to use them.

General

Title here — **Terms**

```
\begin{terms}[]{Title here}

\end{terms}
```

Title here — **Notation**

```
\begin{notation}[]{Title here}

\end{notation}
```

Title here — **Recall**

```
\begin{recall}[]{Title here}

\end{recall}
```

Title here — **Remarks**

```
\begin{remarks}[]{Title here}

\end{remarks}
```

Title here — **Usage**

```
\begin{usage}[]{Title here}

\end{usage}
```

Title here — **Tutorial**

```
\begin{guides}[]{Title here}{Tutorial}
    % You can also change the right title on this one

\end{guides}
```

Title here — **Properties**

```
\begin{properties}[]{Title here}

\end{properties}
```

Title here — **Restrictions**

```
\begin{restrictions}[]{Title here}

\end{restrictions}
```

Title here **Limitations**

```
\begin{limitations}[]{Title here}

\end{limitations}
```

Title here **Intuition**

```
\begin{intuition}[]{Title here}

\end{intuition}
```

Can also be used inline using `\inlineintuition`, which renders **Intuition:**

Counter-enabled

These ones also have two settings, namely, you can change the counter behaviour and the inclusion of subsections in the numbering. See 6.2.

See 6.6.3 for a guide on how to change the current number and how to reference them.

Title here — **Definition 6.1**

```
\begin{definition}[]{Title here}

\end{definition}
```

Title here — **Theorem 6.1**

```
\begin{theorem}[]{Title here}

\end{theorem}
```

Title here — **Lemma 6.1**

```
\begin{lemma}[]{Title here}

\end{lemma}
```

Title here — **Corollary 6.1**

```
\begin{corollary}[]{Title here}

\end{corollary}
```

Title here — **Proposition 6.1**

```
\begin{proposition}[]{Title here}

\end{proposition}
```

Title here — **Fact 6.1**

```
\begin{fact}[]{Title here}

\end{fact}
```

Title here — **Axiom 6.1**

```
\begin{axiom}[]{Title here}

\end{axiom}
```

Title here — **Example 6.1**

```
\begin{example}[]{Title here}

\end{example}
```

Title here — **Remark 6.1**

```
\begin{remark}[]{Title here}

\end{remark}
```

Title here — **Formula 6.1**

```
\begin{formula}[]{Title here}

\end{formula}
```

Language-Specific

Title here — **Conjugation**

```
\begin{conjugation}[]{Title here}

\end{conjugation}
```

Title here — **Forms**

```
\begin{forms}[]{Title here}

\end{forms}
```

Flexible

title — **second title**

This Tcolorbox is flexible and can take any main and secondary title, as well as any colour.

```
\begin{general}[]{title}{second title}{red}

\end{general}
```

6.6.2 Inline & Shortened descriptors

Included in `most` *and up*

See 6.6.3 for a guide on how to change the current number.

Inline

Command	Output
`\inlineex`	**Example 6.2:**
`\inlinedef`	**Definition 6.2:**
`\inlinetheorem`	**Theorem 6.2:**
`\inlinelemma`	**Lemma 6.2:**
`\inlinecorollary`	**Corollary 6.2:**
`\inlineproposition`	**Proposition 6.2:**
`\inlinefact`	**Fact 6.2:**
`\inlineaxiom`	**Axiom 6.2:**
`\inlineremark`	**Remark 6.2:**
`\inlineproof`	**Proof:**

Shortened

Command	Output
`\shortex`	**Ex 6.3:**
`\shortdef`	**D 6.3:**
`\shorttheorem`	**T 6.3:**
`\shortlemma`	**L 6.3:**
`\shortcorollary`	**C 6.3:**
`\shortproposition`	**P 6.3:**
`\shortfact`	**F 6.3:**
`\shortaxiom`	**A 6.3:**
`\shortremark`	**R 6.3:**
`\shortproof`	**Proof:**

Inline (Named)

Command	Output
`\fancyex`{*string*}	**Example 6.4:** *(Exercise)*
`\fancydef`{*string*}	**Definition 6.4:** *(Definition)*
`\fancytheorem`{*string*}	**Theorem 6.4:** *(Theorem)*
`\fancylemma`{*string*}	**Lemma 6.4:** *(Lemma)*
`\fancycorollary`{*string*}	**Corollary 6.4:** *(Corollary)*
`\fancyproposition`{*string*}	**Proposition 6.4:** *(Proposition)*
`\fancyfact`{*string*}	**Fact 6.4:** *(Fact)*
`\fancyaxiom`{*string*}	**Axiom 6.4:** *(Axiom)*
`\fancyremark`{*string*}	**Remark 6.4:** *(Remark)*
`\fancyproof`{*string*}	**Proof:** ***(Proof)***

Shortened (Named)

Command	Output
`\compactex`{*string*}	**Ex 6.5:** *(Exercise)*
`\compactdef`{*string*}	**D 6.5:** *(Definition)*
`\compacttheorem`{*string*}	**T 6.5:** *(Theorem)*
`\compactlemma`{*string*}	**L 6.5:** *(Lemma)*
`\compactcorollary`{*string*}	**C 6.5:** *(Corollary)*
`\compactproposition`{*string*}	**P 6.5:** *(Proposition)*
`\compactfact`{*string*}	**F 6.5:** *(Fact)*
`\compactaxiom`{*string*}	**A 6.5:** *(Axiom)*
`\compactremark`{*string*}	**R 6.5:** *(Remark)*
`\compactproof`{*string*}	**Proof:** ***(Proof)***

6.6.3 Changing the counters

Included in `most` *and up*

Starting from the version of October 18, 2025, you may no longer use `\setcounter` *directly*

You may set the current number for the elements by setting their corresponding counter to the selected number. You can do this using `\setLabelNumber`{*name of the environment*}{*number*}, where you replace *name of the environment* with one of the following: `definition`, `lemma`, `theorem`, `corollary`, `proposition`, `fact`, `formula`, `axiom`, `example`, `remark`

This only applies if you have set `numberingConfig` to 1 (for all of them) and 2 (only for `definitions`). To change the combined numbering, use `\setLabelNumber{all}` to your desired number.

You may also use `\stepLabelNumber`{*name of the environment*} to step the counter by one.

6.6.4 Referencing counters

Included in `most` *and up, introduced on October 18, 2025*

If you wish to reference a counter, you can do so by writing

`\ref{<counter name>:<section>-<subsection>-<subsubsection>-<counter value>}`,

or of course, if you do not label subsections and / or subsubsection, dropping them.

6.7 Tables

Included in `most` *and up*

You can set up nice looking tables using the `booktab` and `tabulary` environments.

```
\begin{tables}{ll}{Left & Right}
    Left content & Right Content\\
\end{tables}
```

This outputs as

Left	Right
Left content	Right Content

This doesn't use the `table` environment, so no captions are possible, as to why there is also

```
\begin{fullTable}{ll}{Left & Right}{This is a caption}
    Left content & Right Content\\
\end{fullTable}
```

This outputs as

Left	Right
Left content	Right Content

Table 1: This is a caption

6.8 Index

Included in `most` *and up*

If you want to use index, add `\prepareIndex` to the preamble. Using `\addIndexBold{`*string*`}`, you can add entries to the index. They are printed in bold typeface in your document. Using `\addIndex{`*string*`}`, you can do the same, but the text remains normal and using `\addIndexItalic{`*string*`}`, it is printed in italics

6.9 Extras

6.9.1 BibTeX

Included in ***recommended*** *and up*

Use `\setupBiber{`*/path/to/your/bib/sources.bib file*`}` in the preamble to prepare, then use `\printbib` to print your bibliography.

To add more sources, simply use bibter's built-in macro `\addbibresource{`*filepath*`}`, which will load your `.bib` file. You need to use that inside the preamble as well.

6.9.2 Glossary

Included in ***recommended*** *and up*

Use `\setupGlossary` in the preamble to prepare, then use the normal glossary commands to add entries to the glossary. When you want to print it, use `\printGlossary`.

6.9.3 Minted

Included in ***full***

Be sure to enable `-shell-escape` for your compiler *(Note: Some newer versions of LaTeX do no longer require this and it is better to keep it turned off for security reasons, if it works without)*!

When `minted` is available via these helpers, you will also have access to the `code` environment:

```
\begin{code}{language}
    //Code goes here
\end{code}
```

This is nothing different than a wrapper for a minted environment with a box drawn around it.

Example (in python):

```
1 def hello_world():
2 print("hello world!")
```

If you want to print code inline, use `\inlinecode{`*string*`}`, which renders to `code`.

Caveat: This is not using the ***verbatim*** *environment due to various limitations of that environment and instead is simply using* `\texttt{}`, so you cannot use this to print LaTeX-commands

7 Troubleshooting

7.1 Visual Errors

7.1.1 Missing headers & footers

You have most likely forgotten about `\startDocument` after `\begin{document}`, or you are using `\setupbarebones` or `\setupexams`.

You can easily distinguish from simply looking at the PDF. If the title is missing too, it's the first one, if it is there, it is most likely the second.

7.1.2 Right box of guides tcolorbox is a weird title

You have most likely forgotten about the second argument. Then, the first letter is removed from the body and used as the title.

7.2 pgfkeys: Don't know mainboxstyle

You have most likely forgotten about `\usetcolorboxes` after `\startDocument`.

7.3 Numbering incorrect after numberingOn

Check the start of the document and make sure you have used `\setNumberingStyle` to set the numbering style and not `\setcounter{numberingConfig}`

7.4 Undefined commands

You have most likely loaded a too small *scope*

7.5 Lots of errors and no compile

You have most likely selected a non-existant *scope* in the `\load` function. You can verify by searching the `<document name>.log` file for any mention of `INVALID CONFIG SPECIFIED, NOTHING LOADED!`

7.6 Minted

`minted` is a syntax highlighting library. It can cause issues when running, as it needs extra configuration for the compiler.

7.6.1 You must invoke LaTeX with the -shell-escape flag

You have set the *scope* to `full`, which loads the `minted` package for code highlighting. You will need to configure your latex compiler to use `-shell-escape` if you want to use it. If you do not plan to use it, simply switch to a smaller *scope*.

7.6.2 You must have pygmentize installed

Your host system is lacking the `pygmentize` package or you have not added the `pip` path to your $PATH. This is also the reason as to why the `-shell-escape` flag has to be set, as minted needs to access external libraries (namely `pygmentize`) to do the syntax highlighting.

7.7 Any other error

Ensure that you are not missing any closing brackets or a math environment is still open. If nothing helps, contact support at `https://support.janishutz.com`