

# L<sup>A</sup>T<sub>E</sub>X Quickstart Guide

COMP 170 Staff

## 1 Introduction

In the world of mathematical and scientific typesetting, L<sup>A</sup>T<sub>E</sub>X is king. L<sup>A</sup>T<sub>E</sub>X is a Turing complete language that compiles to beautiful documents and as a matter of fact, the guide that you are reading at this very moment was written in L<sup>A</sup>T<sub>E</sub>X . We know that learning any new programming language is scary. Thus, we have produced this document to get you writing L<sup>A</sup>T<sub>E</sub>X and creating beautiful PDF documents as fast as possible.

## 2 Your First Document

You can write L<sup>A</sup>T<sub>E</sub>X in any one of your favorite editors and we will show you how to compile and create a PDF later. However, if you want to use a L<sup>A</sup>T<sub>E</sub>X specific IDE for linting, auto-complete etc., you can jump ahead to the “Tools and Compilation” section.

To get started, you can think of L<sup>A</sup>T<sub>E</sub>X as HTML5 for writing technical documents. Guides like this have been written before so we have chosen not to reinvent the wheel. Go to this [website](#) to learn how to write your first document!

If you ever want to typeset a symbol and do not know the L<sup>A</sup>T<sub>E</sub>X command for it, [Detexify](#) is an invaluable resource. Detexify is a website that converts handwritten input to L<sup>A</sup>T<sub>E</sub>X commands.

## 3 Tools and Compilation

In general, you have two general options for writing and compiling L<sup>A</sup>T<sub>E</sub>X . If you want the lowest barrier of entry, we would recommend using an IDE or

a web based editor. If you foresee a future in which you will be writing other technical documents, we recommend writing  $\LaTeX$  in your favorite editor and configuring the appropriate compilation actions (this is often very easy); you will thank yourself later.

### 3.1 From the command line

Just like a C/C++ program,  $\LaTeX$  is compiled. Get your distribution of  $\LaTeX$  for your operating system [here](#). (We use [MikTeX](#) for Windows and [MacTeX](#) for Mac.)

These distributions comes with many programs. You may want to, and should, add the directory where these programs live to your path. One program called `pdflatex` is particularly important; it compiles  $\LaTeX$  into a PDF

With your  $\LaTeX$  document `mydocument.tex` in hand, simply run

```
pdflatex mydocument.tex
```

to produce a PDF called `mydocument.pdf`.

If you want more information about the compilation of  $\LaTeX$  , you can go [here](#).

(For Mac users, there is a guide for troubleshooting [here](#). If you are a Vim or Emacs user, use this to configure your text editor to get from writing  $\LaTeX$  to compilation as fast as possible.)

### 3.2 MacVim + Skim

For people that like using Vim, you can set up MacVim (a GUI version of Vim) to work nicely with  $\LaTeX$  and Skim.

#### Installation

1. Install [Macvim](#). Once installed, link the macvim installation:  

```
ln -s /usr/local/bin/mvim vim
```
2. Install [Skim](#). In the Skim Preferences, go to the Sync tab and under PDF-TeX Sync support, choose “Macvim”.
3. Install [macvim-skim](#).

## Usage

1. Open up a tex file in macvim: `mvim hello.tex`
2. You can compile the tex to PDF using the macvim-vim shortcut. By default, `<leader>t`.
3. Open the PDF in Skim (side by side with macvim)

## Shortcuts

From Macvim:

- `<leader>v`: view current tex file in Skim at current line
- `<leader>p`: rebuild current tex file with `pdflatex -synctex=1`, then activate Skim at current line
- `<leader>m`: Run the command `make` in the current directory, then activate Skim at current line
- `<leader>r`: activate Skim at current line, then bring vim back to the front
- `<leader>t`: `<leader>p` + `<leader>r`: rebuild current file, then bring vim back to the front

From Skim:

`Shift-Command-click` on a line to go to that line in the `.tex` file

## 3.3 Sublime

There is a wonderful free sublime package called [LaTeXTools](#) for  $\text{\LaTeX}$  development. On a Mac, `cmd + b` will compile and open the compiled document on a PDF viewer called Skim (you will need to install Skim separately). Skim will also put a nice red dot on the PDF right where your cursor is in Sublime.

The package will work on Windows, however you will have to configure the plug-in to open some other PDF viewer; Skim is not available on windows.

### 3.4 IDE: TexMaker

IDEs are all the rage these days and [TexMaker](#) is one of our favorites. It comes with auto-complete and, the classic, compile with F5. Some configuration might be required but there are no longer any questions Google (or Bing) [cannot answer](#). (Here is a fun [tidbit](#).)

### 3.5 Web: Overleaf

Overleaf has become one of the most popular places to create  $\text{\LaTeX}$  documents. This will most likely be the option with the lowest barrier to entry. (You will need an Internet connection for this). Go <https://www.overleaf.com> to find out more!

### 3.6 Others

Of course, there are many other development environments and here are some other popular development environments:

- [Tex-Edit](#)
- [TeXworks](#).

There are other plug-ins to help with writing  $\text{\LaTeX}$  in almost any editor you name. For example, you can use the LaTeXTools release for Atom if you so desire, there is [AUCTeX](#) for emacs users, there is [Vim-LaTeX](#) or [LaTeX-Box](#) for vim users.

If you want to immerse yourself and bask in the power of  $\text{\LaTeX}$ , you can check out this [project](#) that was created at Tufts. (Those of you who have taken COMP 40 or 105 might find this familiar).

## 4 Final thoughts

We think that knowing basic  $\text{\LaTeX}$  is an absolutely invaluable skill. At the end of the day there is little value in arguments, thoughts and ideas that are not well documented. Of course, a well written document in whatever form is a triumph in itself but why not make these achievements beautiful as well.

## **Contributors**

Jason Fan, Zach Kirsch, Naomi Zarrilli, Norman Young,