

# A First Set of L<sup>A</sup>T<sub>E</sub>X Packages

Jim Hefferon

T<sub>E</sub>X Users Group Annual Conference 2020-July



# Abstract

This describes a curated list of packages that covers most of what beginners want to do. It seeks to name one package in each area that is capable and reliable.



# Overview

At TUG 2019 I reported on using social media to help understand the needs of today's beginners. Often they just need the right package. I am developing a suitable set.

I kept the document to two sides of a page, aiming to have only one package per problem. Beyond solving the problem, the criteria for a package to be included is that it is in the distributions and largely bug-free. The end document will be in PDF, HTML, and video. The PDF will be on CTAN.

I'm here to solicit feedback.



# Document structure

The document core consists of a few sections classifying areas. Each package name is a hyperlink, with a terse description. There are also a few extra comments in parentheses that often come up in conjunction with the recommendations, and below I have omitted some of these.

Before the list is an introduction. It mentions CTAN. It also mentions using texdoc to read local documentation. Finally, it notes that if a person is writing for a journal or institution then they must see if it has its own package.



# Package set



## Every document

- ▶ To change margins, page size, and orientation, use `geometry`.
- ▶ Get multiple columns with `multicol`.
- ▶ Any document containing significant amounts of mathematics should use the American Mathematical Society's packages `amsmath` and `amssymb`, as well as `amsthm` for producing theorem environments. Don't load `amsmath` directly, instead get it by loading `mathtools`, which adds some useful improvements.
- ▶ You can toss in `microtype`. My eye can't spot the improvements but I appreciate that it means that there are fewer awkward lines.

## Inside the document

- ▶ To tweak lists, use `enumitem`.
- ▶ Enhance captions with `caption`. Control floating environments with `float`. (In particular, if you want an option that overrides automatic float placement and puts something exactly where you ask, this package provides the option ‘H’.)
- ▶ Get hyperlinks and turn references into links with `hyperref`. Make cross-references say ‘Theorem 1.2’ instead of just ‘1.2’ with the one-r-ed `cleveref`. Have URL’s and file paths that can linebreak with `url`.
- ▶ I do code listings with `listings` (although `minted` also has a lot going for it). Make single quotes inside verbatim text come out correctly with `upquote`.

- ▶ For code in Python have a look at [pythontex](#), which, besides showing the code listings, also allows you to execute Python and put the results in your output. Do the same for the *Sage* mathematics software suite with [sagetex](#), and similar systems exist for R, Haskell, and Scheme.
- ▶ There are many package that add table capabilities such as multirow entries and breaking across pages. I most often use [array](#), which lets you define your own columns. To handle units, use [siunitx](#) (which also has a table column type for aligning on a decimal point).
- ▶ To make boxes that are colored or framed, such as boxes for theorems, I use [mdframed](#).
- ▶ Finally, when developing a document I often want some filler text. I use [lipsum](#).



## Graphics and color

- ▶ To include graphics in files, and to do simple manipulation such as resizing, use `graphicx`. Include parts of a PDF document with `pdfpages`. Include video or sound using `media9`.
- ▶ For colors use `xcolor`.
- ▶ To make plots and graphics, I use *Asymptote*, a development of METAPOST with three dimensional constructs. However, many people instead draw graphics inside the document with `TikZ`.

## Front and back matter, headers, footers

- ▶ To style chapter and section titles, use `titlesec`.
- ▶ For page headers and footers, reach for `fancyhdr`.
- ▶ You can tweak the format of tables of contents, lists of figures, etc., with `tocloft`.
- ▶ Write answers to exercises to an external file so you can read them in later with `answers`.
- ▶ I like footnotes at the page bottom so I use `footmisc`.
- ▶ Make an index with `makeindex`.
- ▶ Bibliographies are a thorny area, with lots of strict requirements. CTAN is a big help here since it has many styles for both `BIBTEX` and `biblatex`.

## Special documents

- ▶ Make exams and problem sets with the `exam` class.
- ▶ There are many, many resume and CV packages. Have a look at CTAN's `cv` tag.
- ▶ To make presentations use the `beamer` class.

## Fonts and engines

- ▶ To see options besides the default Computer Modern fonts, visit the [L<sup>A</sup>T<sub>E</sub>X Font Catalogue](#), which includes copy and paste code to make each one work.
- ▶ Beyond that list, you can also use any font that your computer has. To convert L<sup>A</sup>T<sub>E</sub>X source to PDF there are three programs, called engines. Most people use pdfL<sup>A</sup>T<sub>E</sub>X. The XeL<sup>A</sup>T<sub>E</sub>X engine and the LuaL<sup>A</sup>T<sub>E</sub>X engine can leverage the [fontspec](#) package to use your system's fonts.

Take away



## Questions?

- ▶ I have omitted a number of packages that a person may think belongs. Suggested to me were mhchem, physics, and memoir. I also struggled with a few table packages.
- ▶ As I say, I welcome feedback.