Getting Started with LaTeX in the Sagemath Cloud

S.R. Doty

November 7, 2013

LaTeX is a system for creating beautiful documents containing complicated mathematics, tables, diagrams, images, and code. To try it out without any commitment, open a modern browser (not Internet Explorer) and go to

`cloud.sagemath.org.`[1]

Go ahead and sign up for a free account. Once you are logged in, create a new project and select the LaTeX option. This opens a pair of side-by-side panes in the browser. The left pane contains LaTeX commands and the right one shows the "compiled" paper.

There should already be a bare paper skeleton there. Make changes by typing in the left pane. Every time you hit the Save button, the right pane is updated. When you're done editing, click on the Files menu to download your work. If you click on Files now, you will see files with extensions of `synctex.gz`, `pdf`, `log`, `aux`, and `tex`. These are all created by TeX's engine when your paper is compiled; the most important ones are the `tex` file (where the LaTeX input is stored) and the `pdf` file for the converted output. If you have an existing `tex` file you can upload it to the project. Most people learn LaTeX by altering someone else's existing document.

In the Sagemath Cloud, you can easily share your project with one or more collaborators, and they will be able to edit the document, too.

LaTeX is a markup language. You type source commands in one file, which are instructions for what goes where in the document, and then LaTeX converts the code into a PDF document.

Leslie Lamport created the original version of LaTeX based on the earlier TeX project, created in the 1980s by Donald Knuth. Both are open-source and free. You can think of LaTeX as a layer of convenient macros placed on top of TeX; in particular, when you use LaTeX you are also using TeX, and it is TeX's engine that does most of the processing.

All TeX commands begin with a backslash (\). A LaTeX file begins with a `\documentclass` command and always contains a `\begin{document}`—`\end{document}` pair. Anything after `\end{document}` is ignored. Between the `\documentclass` first line and `\begin{document}` is an area called the *preamble*; this is where you put commands with global effects, set options, define your title and author, and other preliminary things. Between `\begin{document}` and `\end{document}` is the *body* of the paper, where you type the paper's content. Anything typed after a `%` symbol is a comment in the file, ignored by TeX.

[1] The Sagemath Cloud™ is a website developed by William Stein of the University of Washington, supporting many types of computational mathematics (including LaTeX, Python, and Sage) in the cloud using a Sage engine running in the background.

To write prose, type it in the body as usual in the left pane. One or more blank lines always indicates a new paragraph. The command `\noindent` before a paragraph suppresses the indentation if its first line. Spacing is automatically calculated by the TeX engine, so you may leave as many blank spaces between words as you like (TeX ignores extra “white space”). To *emphasize* text use the command `\emph{emphasize}`. Bold text is achieved similarly by the `\textbf` command, for instance typing `\textbf{bold}` produces **bold**. The command `\underline{underline}` produces the result underline.

Two types of math are supported: inline and display. Inline math is always placed between a pair of dollar signs, while display math is placed between a pair of double dollar signs. For example, typing `$x^2+y^2=z^2$` produces the inline math equation $x^2 + y^2 = z^2$ while typing `$$x^2+y^2=z^2,$$` produces

$$x^2 + y^2 = z^2,$$

a displayed version of the same equation. Displayed math appears centered on a line by itself. Other ways of producing display math exist. For example, putting math between `\[` and `\]` is equivalent to surrounding it by double dollar signs (so typing `$$x^2+y^2=z^2$$` is equivalent to typing `\[x^2+y^2=z^2\]`). White space in math mode is always ignored by TeX since TeX computes the spacing in math mode (but there are special commands for tweaking that when needed).

LaTeX environments are used for special formatting needs. An environment always starts with \begin{<ename>} and ends with \end{<ename>}, where <ename> is the name of the environment. For example, use the `enumerate` environment to create an indented numbered list. Typing the input:

```
\begin{enumerate}
\item This is the first thing.
\item This is the second thing.
\item This is the third.
\end{enumerate}
```

produces the output

1. This is the first thing.
2. This is the second thing.
3. This is the third.

An unnumbered list (with bullets) is created similarly using the `itemize` environment in the same way.

Theorems, propositions, lemmas, corollaries, definitions, and examples may be formatted using a special environment for that purpose. Such theorem-like environments must be defined in the preamble. For example, after putting the defining command

```
\newtheorem{theorem}{Theorem}
```

in the preamble of this document, I can type something like this in the body

```
\begin{theorem}[Fermat]
There do not exist positive integers $x, y, z$ such that
$x^n+y^n=z^n$ when $n$ is any integer greater than 2.
\end{theorem}
```

which is typeset by LaTeX as follows:

Theorem 1 (Fermat)**.** *There do not exist positive integers x, y, z such that $x^n + y^n = z^n$ when n is any integer greater than* 2*.*

To suppress the (automatic) theorem number, it is necessary to load the `amsthm` package in the preamble and also alter the command defining the theorem environment, as follows:

```
\usepackage{amsthm}
\newtheorem*{theorem*}{Theorem}
```

which should define an unnumbered theorem environment named `theorem*`. Now typing the input:

```
\begin{theorem*}[Fermat]
For any integer $n$ greater than $2$, there do not
exist positive integers $x, y, z$ such that
$x^n+y^n=z^n$.
\end{theorem*}
```

produces the output

Theorem (Fermat)**.** *For any integer n greater than* 2*, there do not exist positive integers x, y, z such that $x^n + y^n = z^n$.*

There are many packages (such as `amsthm`) which enhance the capabilities of LaTeX. Use the `geometry` package to control the margins of the resulting document. Use the `amsmath` and `amssymb` packages to get various convenient extra environments for display math and extra special symbols, respectively. For instance, from amsmath you get the useful `align` and `align*` environments, for creating sequences of aligned displayed equations; this is useful for displaying mathematical derivations.

Computer code is easily rendered in a monospaced font so as to preserved all spacing and endlines, by placing it in a `verbatim` environment. The following Python code was typeset by placing it within `\begin{verbatim}` and `\end{verbatim}`.

```
def hello_world(n):
    """Say hello n times"""
    for j in range(n):
        print("Hello, world!")
```

As you can see, the important indentation was preserved. For fancier ways to display computer code in LaTeX check out the `listings` package, which provides the `lstlisting` environment.

To get started, grab an existing LaTeX file, upload it to the Sagemath Cloud, and start experimenting. You see the effect of your changes as you make them, and you can't break anything, so be brave.

Google search will quickly find answers to most LaTeX questions. In particular, search for LaTeX cheat sheets, which can be very helpful. There are useful online books about LaTeX; one of the better ones is the *Not so Short Introduction to* LaTeX by Tobius Oetiker et al. Printed books are also available if you want to really learn all the myriad ins–and–outs; one of the best is the book *Math into* LaTeX by George Grätzer.

Happy LaTeX-ing!