

Quick Start Guide to \LaTeX , Overleaf, and Mathematics Writing Style

Kenneth M Monks



Chapter 1

A brief introduction to \LaTeX and Overleaf

Created in the late 1970s and early 1980s by Donald Knuth and Leslie Lamport, \LaTeX has become the standard typesetting language used in STEM fields as well as some subdisciplines of psychology, philosophy, political science, and especially linguistics. Overleaf is a cloud-based compiler and shared editing system (similar in many ways to Google Docs) for \LaTeX . Here is a short introduction to the \LaTeX commands, Overleaf functionalities, and writing style principles that you'll need most frequently throughout this course.

1.1 Document structure

Every \LaTeX document begins with a preamble. This must contain a command that declares what type of document you are creating. For example, *article*, *book*, and *exam* are all valid classes of documents. This type is put into curly braces after the word *documentclass* to accomplish this. For example, to write an article we could begin with the command


```
\documentclass{article}
```

which tells the compiler we are creating an article. To start the actual document, one then delineates a *document* environment using

```
\begin{document}
```

The body of your document goes here!

```
\end{document}
```

and anything we want to show up should be placed in that body. Once you have such a document, you can compile it into a pdf, using the **DOWNLOAD PDF** button  in Overleaf or your own local compiler on your machine. Here is a very simple \LaTeX document we could build, which would simply display our foodie observation below.

Your \LaTeX code	What it generates
<pre>\documentclass{article} \begin{document} Gnocchi with marinara constitutes a tomatoed potato. \end{document}</pre>	Gnocchi with marinara constitutes a tomatoed potato.

1.2. MATH MODE, DISPLAY MATH MODE, AND THE ALIGN ENVIRONMENT

For convenience, in the rest of this document we will not show the preamble and the document environment commands, but rather just the code that would go into the body.

Question! 🍷 French fries in ketchup are of course another instance of a tomatoed potato. Can you, however, think of an instance of a potatoed tomato? Show the \LaTeX code that would answer this question.

1.2 Math mode, display math mode, and the align environment

Here we discuss three of the most commonly used capabilities of \LaTeX . *Math mode* is used for displaying small mathematical expressions in a paragraph, *display math mode* is used to display larger mathematical expressions that need to be centered on their own line, and the *align environment* is used to display a sequence of larger mathematical expressions that are connected to each other.

1.2.1 Math mode: show me the money

As we saw above, typing ordinary text into \LaTeX will display words just like any word processor. However, you can toggle *math mode* on and off using dollar signs. This signals to \LaTeX that it should be rendering the expressions you handed it in math mode. For example, it would be **incorrect** to write the sentence below.

Your \LaTeX code	What it generates
In the early 1600s, Ren\'e Descartes introduced the convention of using the letters x,y , and z as variable names.	In the early 1600s, René Descartes introduced the convention of using the letters x,y , and z as variable names.

In this instance, \LaTeX is formatting our variables as if they are just letters appearing in words and not mathematics expressions. (Also notice the backslash-apostrophe creates the accent mark on the next character!) Instead, we should write it as follows.

Your \LaTeX code	What it generates
In the early 1600s, Ren\'e Descartes introduced the convention of using the letters $\$x,y,\$$ and $\$z\$$ as variable names.	In the early 1600s, René Descartes introduced the convention of using the letters x,y , and z as variable names.

Question! 🍷 Why would it be wrong to format the sentence above using the code written below?

In the early 1600s, Ren\'e Descartes introduced the convention of using the letters $\$x,y$, and $z\$$ as variable names?

1.2.2 Display math mode: show me the money money

If you have a larger, more complicated mathematical expression that needs its own space in the middle of the page rather than appearing inline with other text, it is time to use *display math mode*. This is toggled by double-dollar signs as shown below.

Your L ^A T _E X code	What it generates
<p>The quartic polynomial</p> $x^4 - 4x^3 + 6x^2 - 4x + 1$ <p>is most easily factored by recognizing the coefficients from Pascal's Triangle, producing the factorization</p> $x^4 - 4x^3 + 6x^2 - 4x + 1 = (x - 1)^4.$	<p>The quartic polynomial</p> $x^4 - 4x^3 + 6x^2 - 4x + 1$ <p>is most easily factored by recognizing the coefficients from Pascal's Triangle, producing the factorization</p> $x^4 - 4x^3 + 6x^2 - 4x + 1 = (x - 1)^4.$

Question! ♣ One of the most common expressions in mathematics is of course the beloved summation, which in L^AT_EX is called `\sum`. If you use the code

```
\sum_{n=0}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}
```

in math mode vs in display math mode, how does it look different?

1.2.3 Align environment

If you have a sequence of equations or expressions which you would like to display as connected to one another in a sequence down the page, an *align environment* is likely the right choice. In such an environment, the ampersand marks the alignment character, and a double backslash represents a line break. For example, here is another way we could typeset the factorization of the quartic above.

Your L^AT_EX code

```
The quartic polynomial  $x^4 - 4x^3 + 6x^2 - 4x + 1$  can be factored by grouping as follows:
\begin{align*}
x^4 - 4x^3 + 6x^2 - 4x + 1 &= x^4 - 2x^3 - 2x^3 + x^2 + 4x^2 + x^2 - 2x - 2x + 1 \\
&= \left(x^4 - 2x^3 + x^2\right) + \left(-2x^3 + 4x^2 - 2x\right) + \left(x^2 - 2x + 1\right) \\
&= x^2\left(x^2 - 2x + 1\right) - 2x\left(x^2 - 2x + 1\right) + 1\left(x^2 - 2x + 1\right) \\
&= \left(x^2 - 2x + 1\right)\left(x^2 - 2x + 1\right) \\
&= \left(x^2 - 2x + 1\right)^2 \\
&= \left(\left(x - 1\right)^2\right)^2 \\
&= \left(x - 1\right)^4.
\end{align*}
```

What it generates

The quartic polynomial

$$x^4 - 4x^3 + 6x^2 - 4x + 1$$

can be factored by grouping as follows:

1.3. MORE COMPLICATED MATHEMATICAL EXPRESSIONS

$$\begin{aligned}
 x^4 - 4x^3 + 6x^2 - 4x + 1 &= x^4 - 2x^3 - 2x^3 + x^2 + 4x^2 + x^2 - 2x - 2x + 1 \\
 &= (x^4 - 2x^3 + x^2) + (-2x^3 + 4x^2 - 2x) + (x^2 - 2x + 1) \\
 &= x^2(x^2 - 2x + 1) - 2x(x^2 - 2x + 1) + 1(x^2 - 2x + 1) \\
 &= (x^2 - 2x + 1)(x^2 - 2x + 1) \\
 &= (x^2 - 2x + 1)^2 \\
 &= \left((x - 1)^2\right)^2 \\
 &= (x - 1)^4.
 \end{aligned}$$

Notice above the parenthesis all have the words *left* and *right* attached to them; we will discuss this further in the following section!

Question! 🐛 Notice the commands that began and ended the align environment each had an asterisk. What happens if the asterisks are removed?

1.3 More complicated mathematical expressions

1.3.1 Fractions and binomial coefficients

Fractions and binomial coefficients are generated using the commands `\frac{...}{...}` and `\binom{...}{...}`, respectively. Whatever expression is placed in the first set of curly braces goes on top, and the expression in the second set goes on bottom. For example, the code below would typeset the formula for the binomial coefficients.

Your \LaTeX code	What it generates
<code>\binom{n}{k}=\frac{n!}{k!(n-k)!}</code>	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$

1.3.2 Stretchy delimiters

The standard symbols for parenthesis, absolute value bars, and curly braces will automatically grow to fit the size of the expression inside if entered correctly in \LaTeX . Compare the code for parentheses and generated expressions below, shown with and without stretchiness. The key is adding the words *left* and *right* before the corresponding parentheses.

Parentheses!	Your \LaTeX code	What it generates
Bad:	<code>x^2 \cdot (1-e^{x^2+1})</code>	$x^2 \cdot (1 - e^{x^2+1})$
Good:	<code>x^2 \cdot \left(1- e^{x^2+1} \right)</code>	$x^2 \cdot \left(1 - e^{x^2+1}\right)$

Because \LaTeX uses the vertical bar as a special character for other purposes within its language, absolute values are a bit stranger to typeset correctly. They involve in fact not using the `|` key at all, but rather the words *lvert* and *rvert*.

Absolute value	Your L ^A T _E X code	What it generates
Bad:	<code>\ln x^3+y^3 </code>	$\ln x^3 + y^3 $
Good:	<code>\ln\left\lvert x^3+y^3\right\rvert</code>	$\ln x^3 + y^3 $

Likewise, set brackets are used in L^AT_EX to denote an input to a function and as a grouping symbol. Thus, to properly format curly braces, it is best to use the words *lbrace* and *rbrace* and not type the curly brace itself at all.

Set brackets	Your L ^A T _E X code	What it generates
Bad:	<code>\{e^{Cx^2-1}:C\in\mathbb{R}\}</code>	$\{e^{Cx^2-1} : C \in \mathbb{R}\}$
Good:	<code>\left\lbrace e^{Cx^2-1}:C\in\mathbb{R}\right\rbrace</code>	$\left\{e^{Cx^2-1} : C \in \mathbb{R}\right\}$

Question! 🐛 We mention above that curly braces are often used as a grouping symbol. The following example shows more specifically what is meant by that. Try out the following three ways of raising x to the tenth power:

- `x^10`
- `$x^(10)$`
- `x^{10}`

Which one or ones are correct? Which are garbage?

1.3.3 Vectors and matrices

Stretchy square brackets are usually needed in the case of vectors and matrices. This comes up so often that L^AT_EX has a built-in environment called *bmatrix* just for this purpose. Much like align environments, ampersands denote columns to be lined up, whereas double backslash marks the start of a new row.

Your L ^A T _E X code	What it generates
<code>\begin{bmatrix} 3\sqrt{3} \\ 3 \\ 8 \end{bmatrix}</code>	$\begin{bmatrix} 3\sqrt{3} \\ 3 \\ 8 \end{bmatrix}$
<code>\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}</code>	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

1.3.4 Lists of objects

In L^AT_EX, ellipses should never be typed out by repeatedly typing periods. Rather, there are special commands *cdots* and *ldots* for centered dots and lowered dots. The centered dots should be used whenever the center of mass of the adjacent symbol is at the halfway mark; the lowered dots should be used whenever the center of mass is below the halfway mark.

1.4. LISTS: ITEMIZE AND ENUMERATE

Ellipses	Your \LaTeX code	What it generates
Bad:	<code>1+2+3+...+n=\frac{n(n+1)}{2}</code>	$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$
Slightly Better:	<code>1+2+3+ \ldots +n=\frac{n(n+1)}{2}</code>	$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$
Bliss:	<code>1+2+3+ \cdots +n=\frac{n(n+1)}{2}</code>	$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$

Question! 🐛 Why would the following code below be evil and smell bad? How should it be fixed?

```
$a_1,a_2,a_3,\cdots,a_n$
```

1.3.5 Piecewise functions

There is a very nice built in environment called *cases* which handles piecewise functions beautifully. Similar to the notation for vectors and matrices, the ampersand moves the expression to a new column, while the double backslash moves to a new row. The example below shows how you could use this to typeset the definition of the absolute value function.

Your \LaTeX code	What it generates
<pre><code>\lvert x\rvert = \begin{cases} x, & \text{if } x \geq 0 \\ -x, & \text{if } x < 0. \end{cases}</code></pre>	$ x = \begin{cases} x, & \text{if } x \geq 0; \\ -x, & \text{if } x < 0. \end{cases}$

Question! 🐛 What would go wrong in the output above if we did not include those two `text` commands?

1.4 Lists: itemize and enumerate

If you want to create a bulleted list (that simply has markers but is not lettered or numbered in any sequential way), use the *itemize* environment. If you want to create a numbered or lettered list, use the *enumerate* environment instead. Both of these environments use the command *item* to mark a new item in the list. Both of these environments can also be nested to create sublists. Note that the `align` environment automatically puts everything inside into math mode; for *itemize* and *enumerate* everything is in text mode and must be put in math mode manually using dollar signs if needed. It is tradition to indent the items inside such an environment for readability of your code, though such whitespace is actually ignored by the compiler. (which was basically the default in the pre-Python era)

Your \LaTeX code

```
\begin{itemize}
  \item {\bf Sines and Cosines:} $\cos^2(\theta)+\sin^2(\theta)=1$
  \item {\bf Tangents and Secants:} $1+\tan^2(\theta)=\sec^2(\theta)$
  \item {\bf Cotangents and Cosecants:} $\cot^2(\theta)+1=\csc^2(\theta)$
\end{itemize}
```

What it generates

- **Sines and Cosines:** $\cos^2(\theta) + \sin^2(\theta) = 1$
 - **Tangents and Secants:** $1 + \tan^2(\theta) = \sec^2(\theta)$
 - **Cotangents and Cosecants:** $\cot^2(\theta) + 1 = \csc^2(\theta)$
-

The `itemize` environment will be how we format our discussions in the *Comments* sections below each problem. If you are replying to someone above, you create a sublist; if you are posting a new comment then you create a new item.

Question! 🍷 How does the above environment change if you use *enumerate* instead of *itemize*?

1.5 Including graphics

Fancy tools for generating graphs within L^AT_EX itself exist such as TikZ and Asymptote. You are of course welcome to look into these if you choose, but these have a bit of a learning curve. For our purposes, it will likely be easiest to use an external graphing utility to create the image you need and then just import the image into your Overleaf project.

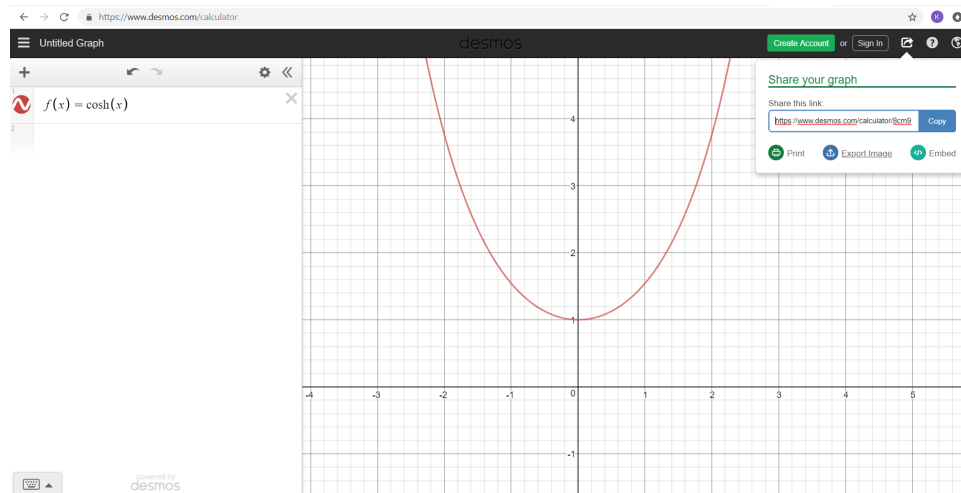
By default, L^AT_EX does not have graphics capabilities built in. You need to load a package called *graphicx* in order to get that functionality by including the command

```
\usepackage{graphicx}
```

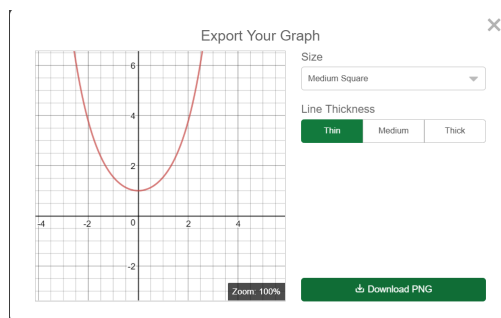
in your preamble (before your document environment starts). Once you have this package, you'll be able to include graphics using the steps detailed below.


1. **Create the graphic:** Start by creating the graphic in your favorite external graphing utility. For our course, particularly useful ones include the following:
 - **Desmos:** Great for plotting graphs of functions, their derivatives, and parametric curves. Super easy interface and low learning curve.
<https://www.desmos.com/>
 - **Geogebra:** Very user-friendly 2D and 3D plotter. Also has a fantastic ruler-and-compass construction application.
<https://www.geogebra.org/>
 - **Darryl Nester's Slope Fields and Phase Planes:** To easily visualize vector fields and see solutions to differential equations, you can't beat this.
<https://www.bluffton.edu/homepages/facstaff/nesterd/java/slopefields.html>
2. **Export and Save:** From that utility, save the graphic locally on your computer. Here we demo Desmos because *demo* is a substring of *desmos*. First, click the SHARE button in the upper right.

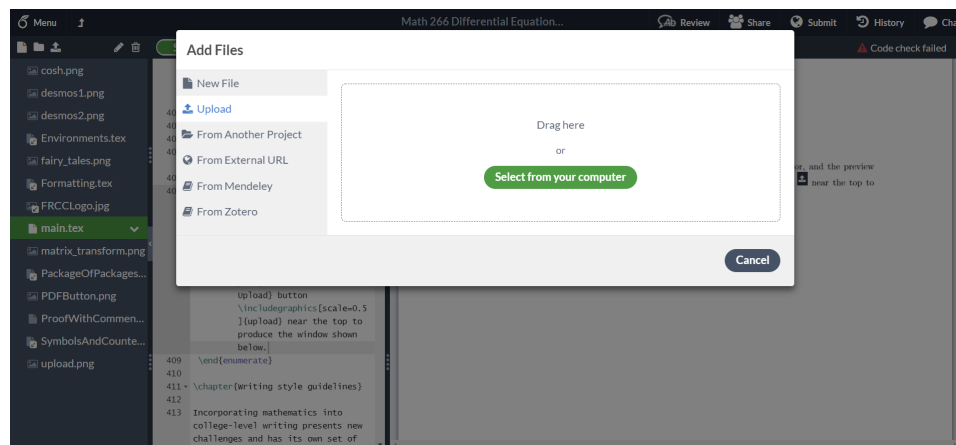
1.5. INCLUDING GRAPHICS



Then, use the **DOWNLOAD PNG** button to create a local copy on your computer. Rename it as you like. Here we call it *cosh.png* since it is a hyperbolic cosine graph.



3. **Upload to Overleaf:** Overleaf has three panes, the file-tree, the code editor, and the preview window, from left to right. In the leftmost pane, click the **UPLOAD** button  near the top to produce the window shown below.



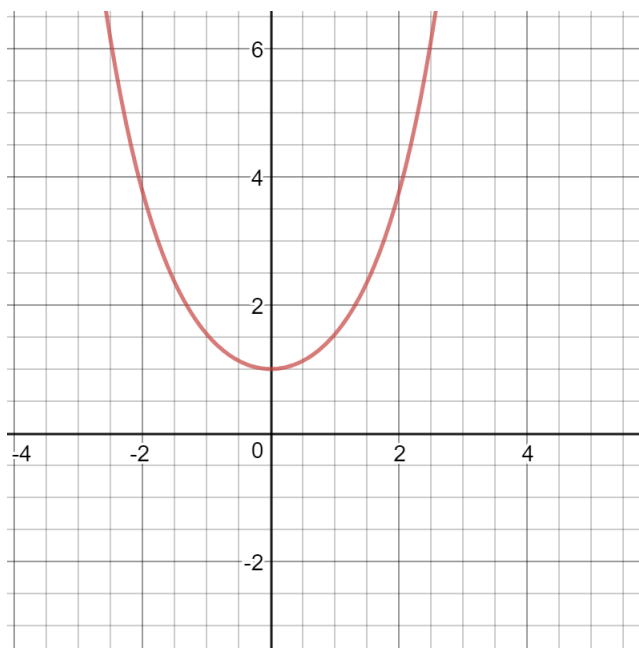
4. Use the ***includegraphics*** command: Wherever you would like the image to appear in your document, use the command

```
\includegraphics[scale=number]{filename}
```

where the name of your file is between the curly braces, and `scale` is equal to whatever factor you would like to resize your image by. It is often desirable to put your image in a `center` environment just to get it centered on the page. For example, the code

```
\begin{center} \includegraphics[scale=0.3]{cosh} \end{center}
```

produces the image below in our document.



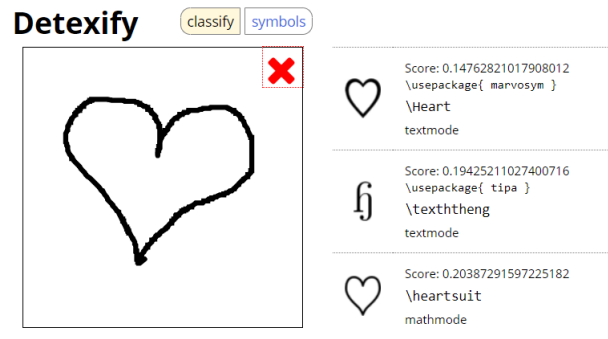
1.6 How to go further

1.6.1 Additional resources

How to best investigate further L^AT_EX functionality depends a lot on what you are looking for. If you are looking for...

- ...a single special character. Try the **Detexify** website or mobile app. It will allow you to draw a character on the touchscreen or with a mouse, and it will take its best guess at what the code is for what you drew. This is not good at telling you how to build more complicated expressions, but for a single character it is great. See my not-so-great drawing on the left, with its guesses as to what I mean on the right!

1.6. HOW TO GO FURTHER



<http://detexify.kirelabs.org/>

- **... a more complicated expression.** Say there is some bigger operation you are trying to perform, like building fancy tables, captioning images, building an autonumbered index, etc. Your best bet here is to type a phrase describing what you want to do into **StackExchange**. Chances are, whatever question you're asking has already been asked by someone else on StackExchange, and someone has answered it.

<https://tex.stackexchange.com/>

- **... more general information.** The \LaTeX wikibook is very well-written and usually quite easy to find what you are looking for. It provides great documentation for the language.

<https://en.wikibooks.org/wiki/LaTeX>

- **... a \LaTeX Sandbox!** There is a browser-based application called *LaTeXNotes* with fast rendering and an easily searchable database of \LaTeX commands. It is useful for quickly composing a component of a larger document which could then be copy-pasted into our Overleaf homework assignments.

<https://latexnotes.surge.sh>

Be aware that any of the code or suggestions you find in the above sources may require loading some extra packages for it to work properly (much like how we loaded *graphicx* above). Pay attention to what packages are in the preamble of sample code you find.

Question! 🐛 Use one of the above suggested resources to find the \LaTeX code for a smiley face. What is the code and how did you find it?

1.6.2 Customizing \LaTeX

One of the most powerful and attractive features of \LaTeX is the ability to create custom commands and shortcuts. This is easily accomplished by writing *newcommand* lines in your preamble. How to accomplish this is most easily seen in an example.

Suppose we were using a lot of absolute values in our document, and we were sick and tired of typing

```
\left\lvert soon-to-be-nonnegative \right\rvert
```

each and every time. We could define a function called *abs* by typing

```
\newcommand{\abs}[1]{\left\lvert #1 \right\rvert}
```

into our preamble. The first argument in squiggles specifies the name of our new command, *abs*, the second argument in square brackets specifies the number of arguments we desire, just 1, and the third expression in squiggles specifies what we want our function to be a shortcut for. The “#1” represents where we want argument number 1 to go in our code. Armed with our new fancy command, we could write the following little program.

Your L ^A T _E X code	What it generates
<pre data-bbox="313 474 997 726"> \documentclass{article} \usepackage{amsmath} \newcommand{\abs}[1]{\left\lvert #1 \right\rvert} \begin{document} \$\$\sqrt{x^2}=\abs{x}\$\$ \end{document} </pre>	$\sqrt{x^2} = x $

This of course is not worth it if you were only going to use this command once, but in a longer writeup it is a great way to customize your L^AT_EX document. It is one of the beautiful aspects of using L^AT_EX. You can teach it new commands to make it the most efficient tool possible for your own purposes.

Chapter 2

Writing style guidelines

Incorporating mathematics into college-level writing presents new challenges and has its own set of agreed upon conventions, much like APA or MLA apply to many disciplines. Here we list a set of commonly followed conventions in addition to the typesetting guidelines listed above.

2.1 Function names are special characters

Never typeset famous function names in italics (even in math mode). Rather, famous function names should always be typeset using a special command starting with a backslash. Very common ones are built in; one can custom build others as needed.

Special functions	Your \LaTeX code	What it generates
Bad:	<code>\cos</code> , <code>\sin</code> , <code>\arctan</code> , <code>\ln</code> , <code>\det</code>	<i>cos, sin, arctan, ln, det</i>
Good:	<code>\cos</code> , <code>\sin</code> , <code>\arctan</code> , <code>\ln</code> , <code>\det</code>	cos, sin, arctan, ln, det

2.2 Mathematics expressions are always contained as part of a grammatically correct sentence

Often in earlier courses where you may be focused solely on calculating some answer, your mathematics writing may have consisted of mathematical expressions written on top of one another. This is very commonly used in scratchwork or chalkboard work, as it is a short way to turn one expression into another equivalent expression. However, just like college-level writing in any other discipline, all mathematics expressions (except for objects like tables, graphs, or appendices, and a few exceptions like agreed-upon formatting conventions such as polynomial long division or matrix row reduction) need to be part of a grammatically correct sentence. As a warning, it will take some time to get used to this style of mathematics writing if you haven't done it before.

Here is an example of how you could take math that is written with a bad case of floating blob disease and transform it into a proper writeup where everything is part of a complete sentence.

Mathematics with floating blob disease:

Solve the differential equation for y .

$$\begin{aligned}\frac{dy}{dt} &= 2y \\ \frac{1}{y} \frac{dy}{dt} &= 2 \\ \int \frac{1}{y} \frac{dy}{dt} dt &= \int 2 dt \\ \ln |y| &= 2t + C_1 \\ y &= \pm e^{2t+C_1} \\ y &= \pm e^{C_1} e^{2t} \\ y &= C e^{2t}\end{aligned}$$

Though this is how you might write such a calculation in your scratchwork on a test or on the chalkboard, this would not be an acceptable way to typeset mathematics. In the writeup above, each mathematics expression is just floating above the other, and none of them live in sentences. Here is one way (of many possible) you could rewrite it to get it up to modern standards of mathematics writing.

Mathematics without floating blob disease:

We wish to solve the differential equation

$$\frac{dy}{dt} = 2y$$

for y . To begin, we divide both sides by y and integrate both sides with respect to t , producing the equation

$$\int \frac{1}{y} \frac{dy}{dt} dt = \int 2 dt.$$

The integrand on the left-hand side is easily recognized as the derivative of $\ln |y|$ with the chain rule applied to the inner function y , while the right-hand side is the integral of the constant function 2. Evaluating each integral produces the relationship

$$\ln |y| = 2t + C_1.$$

At this point, all the calculus is complete and we need only to algebraically solve for y . Solving for y and then cleaning up our constant of integration with the substitution $C = \pm e^{C_1}$ results in the following solution:

$$y = \pm e^{2t+C_1} = \pm e^{C_1} e^{2t} = C e^{2t}.$$

Notice in particular how our sentences end with periods even if they happen to end in a mathematics expression.

Question! 🐛 What would be wrong with the following code?

```
Evaluating each integral produces the relationship

$$\ln|\left|y\right|=2t+C_1$$
.
```

Question! 🐛 In the good writeup above, some expressions were in math mode and some were in display math mode. How do you think the author decided which to use when?

2.3 Normal rules of grammar still apply

All of the usual rules of writing grammatically correct sentences in college-level writing still apply. Commas, colons, and apostrophes in particular are very useful but also easy to misuse. If you type *it's* make sure you mean *it is*, if you type *you're* make sure you mean *you are*, and if you type *let's* make sure you mean *let us*. For other less common cases, I list a few helpful resources below.

- **Mignon Fogarty, Grammar Girl.** Her short articles and podcasts provide great practical advice.
 - **Commas!** <https://www.quickanddirtytips.com/education/grammar/where-do-i-use-commas>
 - **Colons!** <https://www.quickanddirtytips.com/education/grammar/colons>
 - **Apostrophes!** <https://www.quickanddirtytips.com/education/grammar/apostrophes>
- **Purdue OWL.** The Online Writing Lab from Purdue University has become a widely used and respected source on writing. It has videos and notes providing information and exercises on everything from the mechanics of proper grammar to how to keep your audience in mind while writing.
https://owl.purdue.edu/owl/general_writing/index.html

Question 🦉 Any time you use a colon, what word does Mignon Fogarty suggest it should be interchangeable for? That is, a colon is grammatically correct (at least in Chicago style) if and only if it could be replaced by the word _____.

2.4 Never start a sentence with a math symbol

Mathematics writing becomes hard to parse if you allow yourself to start sentences with mathematics symbols. These are usually easily fixed by appositives; simply name the type of object that you wanted to start the sentence with.

Incorrectly starting a sentence with a math symbol:

$f(t) = t$ has a Laplace transform of $1/s^2$.

Fixing it with an appositive:

The function $f(t) = t$ has a Laplace transform of $1/s^2$.

2.5 U-shaped proofs are evil

One common pattern of argument in earlier courses (especially common in trigonometry) is to verify an equation by starting with the equation and then doing things to both sides until you produce something clearly true, like $0 = 0$. This is technically circular reasoning, as you are starting with what is really supposed to be your conclusion.

To see this in an example, consider the two following arguments, both using the fact that $\cos^2(\theta) + \sin^2(\theta) = 1$ to prove that $\tan^2(\theta) + 1 = \sec^2(\theta)$.

Argument 1: U-shaped proof. (Egad! Bad. Sad.)

We wish to show that $\tan^2(\theta) + 1 = \sec^2(\theta)$. We start with this identity and then manipulate both sides until we get something true.

$$\begin{aligned}\tan^2(\theta) + 1 &= \sec^2(\theta) \\ \frac{\sin^2(\theta)}{\cos^2(\theta)} + \frac{\cos^2(\theta)}{\cos^2(\theta)} &= \frac{1}{\cos^2(\theta)}\end{aligned}$$

Now multiply both sides by $\cos^2(\theta)$,

$$\cos^2(\theta) + \sin^2(\theta) = 1$$

which is true, so we have proven the identity.

This is how identity “proofs” are often taught in trig; in that context it is fine to work this way as you are basically just trying to get used to applying the identities and doing algebra with trig functions. But as far as an argument goes, the logical flow is awful. What you’re really saying is that the first equality is equivalent to the second which is equivalent to the third, and since the third is true, then so is the second, and so is the first. It’s not wrong, but it’s really convoluted. Also it’s considered bad form because it really looks like your starting point of the whole proof is the very thing that you’re trying to prove.

How to fix this? Just start with the left-hand side of that identity you’re trying to prove, and then continually manipulate it until you reach the right hand side. It’s not much work to take a U-shaped proof and turn it into one of these, and it is considered *much* better style. This is called a *transitive chain* and is the right way to rewrite a U-shaped proof (not to mention fix the bad case of floating blob disease happening above). In a transitive chain, we use an *align* environment to say the first expression equals the second which equals the third and so on down to the bottom; at the end we conclude the first equals the last since equality is a transitive relation.

Argument 2: Transitive chain. (Glad! Rad. Radian.)

We wish to show that $\tan^2(\theta) + 1 = \sec^2(\theta)$. We start with the left-hand side and manipulate it using the Pythagorean identity for sines and cosines until we obtain the right-hand side as follows:

$$\begin{aligned}\tan^2(\theta) + 1 &= \frac{\sin^2(\theta)}{\cos^2(\theta)} + \frac{\cos^2(\theta)}{\cos^2(\theta)} \\ &= \frac{\sin^2(\theta) + \cos^2(\theta)}{\cos^2(\theta)} \\ &= \frac{1}{\cos^2(\theta)} \\ &= \sec^2(\theta).\end{aligned}$$

Thus, we have established that $\tan^2(\theta) + 1 = \sec^2(\theta)$.

Chapter 3

Table of common symbols and commands

CHAPTER 3. TABLE OF COMMON SYMBOLS AND COMMANDS

	Your \LaTeX code	What it generates
Multiplication	<code>a \cdot b</code>	$a \cdot b$
Exponentiation	<code>a^b</code>	a^b
Plus or minus	<code>a \pm b</code>	$a \pm b$
Radicals	<code>\sqrt[n]{x}</code>	$\sqrt[n]{x}$
Fractions	<code>\frac{a}{b}</code>	$\frac{a}{b}$
Binomial coefficients	<code>\binom{n}{m}</code>	$\binom{n}{m}$
Subscripts	<code>a_n</code>	a_n
Vectors, single symbol	<code>\vec{x}</code>	\vec{x}
Vectors, component form	<code>\begin{bmatrix} x \\ y \end{bmatrix}</code>	$\begin{bmatrix} x \\ y \end{bmatrix}$
Matrices	<code>\begin{bmatrix} a & b \\ c & d \end{bmatrix}</code>	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$
Summations	<code>\sum_{n=j}^k a_n</code>	$\sum_{n=j}^k a_n$
Integrals	<code>\int_{x=a}^{x=b} f(x) dx</code>	$\int_{x=a}^{x=b} f(x) dx$
Stretchy parentheses	<code>\left(math^{mathitymath} \right)</code>	$(math^{mathitymath})$
Stretchy absolute value	<code>\left\lvert math^{mathitymath} \right\rvert</code>	$ math^{mathitymath} $
Stretchy squiggles	<code>\left\{ math^{mathitymath} \right\}</code>	$\{math^{mathitymath}\}$
Famous sets of numbers	<code>\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}</code>	$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$
Common Greek letters	<code>\theta, \delta, \epsilon, \lambda, \pi</code>	$\theta, \delta, \epsilon, \lambda, \pi$

Question! 🐾 What symbol does the Colorado Department of Transportation use for multiplication?

Chapter 4

Why?

Why am I structuring the course this way? Why I am requiring use of this system? Wouldn't good old pencil and paper be easier and better? Well, the answer is at first maybe yes, but in the long run, definitely no. Although there will likely be a slow or frustrating period in the beginning, once you get over that hump it will help you fly for the rest of your STEM career. Here are my objectives with regards to structuring the course around this homework system:

- **Better feedback from me.** I want all of you to have more feedback and better quality feedback from me on your homeworks. Pen and paper is just so slow for me to write anything substantial, especially when there are similar mistakes on 15 different assignments, and inevitably the one on the bottom of the pile is getting shorter and sloppier feedback than the ones on the top of the pile.
- **Better feedback from each other.** Here at BCC, we're not at CU where everyone is a 19-year old full-time student living in the Engineering dorm, and you can always just go knock on the door of the twelve people down the hall who have the same homework assignment to ask a question. So, I wanted to create this shared space where you do have access to your peers (of all different ages and work schedules and living situations) 24/7 and can collaborate, rather than feeling isolated when working on homework.
- **Research-backed benefits of active learning.** There are volumes of research supporting active learning as a more effective learning modality than listening to lecture alone. Studies show that comprehension and retention from listening to lecture alone is fairly low compared to when it is coupled with higher-level processes like writing, formulating questions based on reading, and sharing ideas with collaborators. One problem is that in higher-level courses, we sometimes don't have time to do much more than state our definitions and work out one big example in class, so with this online system we can still have good thorough demonstrations of the mathematics in class, with the added benefits of active learning in the homework.
- **Forum for sharing resources.** There are a million and seven free mathematics resources out there, but it's a minefield riddled with lots of misleading and poor quality time-suckers in the higher levels of undergraduate mathematics. I have already shared a few resources in each assignment that I have found very helpful; if you find any that you like, you can share them there as well!
- **Getting your technical writing up to modern standards of professionalism and academic maturity.** Whether you end up working in government, industry, or academia, technical documents are never written without proper sentence structure, and it is best to have your first round of getting used to that on low-stakes homeworks in a class instead of in the workplace for the first time. You do not want to end up feeling like this poor job interview applicant did at two minutes in.

<https://www.youtube.com/watch?v=dY2mRM4i6tY>

- **Practice incorporating feedback in your technical writing.** This style of working much more accurately simulates what will happen down the road. I can't think of even a single time in my career where I just wrote an important technical document and then didn't have to thoughtfully incorporate some feedback from my department chair, a dean, a fellow faculty member, an adjunct, a student, a grant-reviewer, my advisor, a journal referee, a journal editor, etc. This doesn't mean that you always make every change that is suggested to you, but you should incorporate it thoughtfully. Maybe you disagree with the change that someone suggested, but you see why it caught their eye and it still brings to light some third alternative regarding how you can improve your document. It is an incredibly important skill that takes time and practice to develop.
- **Practice reading technical documents critically and giving others constructive feedback.** The assumption in most undergraduate courses is that every document you read is perfect and correct. The fancy expensive textbook is from some giant publishing company with an army of content experts and editors proofreading it. However, this Candyland fantasy quickly vanishes as you enter a more specialized niche in whatever field you work in. Often there aren't that many people worldwide working on it and a lot of the ideas are too new to even know what the gold-plated final form of them should be. In roles as a colleague, supervisor, or researcher, you will frequently have to read technical writing and provide the author with feedback on the writing or make your own judgement as to what seems correct or incorrect (and contact the author or others in your field with these questions). Like anything worthwhile, this will take time and effort to develop this skill.
- **Having you leave the course knowing L^AT_EX.** This is an extremely good tool to carry forward with you for the following reasons:
 - **Secret handshake in the STEM workplace.** In technical fields these days, there is a bit of an unspoken culture that if you go in with a document not prepared in L^AT_EX, the first impression is automatically a negative one, and if it is in L^AT_EX, it's automatically positive. Preparing technical docs these days in Microsoft Word would be kind of the same as if a professional wedding photographer did all their editing in Instagram instead of Photoshop. Yes of course Instagram is way easier to use and Photoshop has a steep learning curve, but if you are going to do real work in any field, you need to have more control over your own projects and not have a super hand-holdy program doing all sorts of auto- this and auto- that for you.
 - **Common requirement in upper level coursework at universities.** Lots of professors at CU, CSU, etc require assignments or lab reports to be done in L^AT_EX. For example, I know the main Circuits professor at CU Boulder does. (And as you might expect, they will not provide scaffolding like I am doing here but rather just say "do this".) This way as you're dealing with all the other challenges of transfer, you at least don't have to also be trying to use L^AT_EX for the first time ever.
 - **Common requirement in journals.** As you try to do research in a STEM field and publish in paper or electronic journals, you will find that most do not even accept submissions in any format other than .TEX files.
 - **Perfect introduction to coding.** If you don't have programming experience, it is the perfect introductory language because the language is very simple compared to say C or Java. But, it at least gets you used to some of the basic syntax of how compiled languages look and how to use them. Historically, L^AT_EX was one of the early languages that served as the basis for HTML, so it's a great launching point for learning other languages (and realistically every STEM professional needs to know at least a bit of coding these days).

Hopefully it is clear that I built this to not only serve as a more effective (and fun!) system for learning the content of this course, but also to set you up for success in broader ways for further STEM adventures!

Question! 🖱️ Visit the url given below:

<https://www.frontiersin.org/journals/psychology/sections/quantitative-psychology-and-measurement#submission-checklist>

What website is this? What purpose does it serve, and for what entity? What two submission formats do they mention in name are acceptable for their journal? Which one do you think they actually prefer and why?

Question! 🖱️ Find another journal of a STEM field you are interested in. What formats do they accept for submissions?