

L^AT_EX for Absolute Beginners: A Short Course

PART ONE

Prepared for the Star Lab by C. Basu*

Last updated: 03 October 2014

Contents

1	An Introduction to Using L^AT_EX	2
1.1	What is L ^A T _E X and why should I use it?	2
1.2	Where can I find L ^A T _E X?	2
1.3	Starter Exercise	3
2	Basic Content in L^AT_EX	4
2.1	Characters & Commands	4
2.2	Whitespace and Spacing	5
2.3	Structure of an Input File	6
3	Full Documents in L^AT_EX	8
3.1	Review Exercise I	8
3.2	Typesetting Text	8
3.3	Typesetting Math	14
3.4	Common Errors and Debugging	20
A	Appendix	21

*This document was prepared by the author for the incoming graduate student cohort at the University of Rochester, and adapted from previous versions of the course taught by Jonathan Olmsted, Dave Armstrong and Arthur Spirling.

1 An Introduction to Using L^AT_EX

1.1 What is L^AT_EX and why should I use it?

L^AT_EX is:

- pronounced “Lay-teck”, not “Latex”.
- *not* a WYSIWYG (“what you see is what you get”) system.
- a document processing language that is particularly useful for typesetting mathematics. For example, you can (without too much elbow grease) produce equations that look like this:

$$\sum_{i=i}^N \alpha_i = \frac{\tilde{\pi}(\eta_L^X + 2\psi\eta_L^Y - \eta_L^Y) + (1 - 2\psi\eta_L^Y + \eta_L^Y)^{\sin\theta}}{\sqrt{5(1 - 2\psi\eta_L^Y + \eta_L^Y)}} \neq 0 \forall \psi$$

- designed with scientific articles and presentations in mind, and therefore handles statistical output and formats tables and figures with ease.
- an open software which creates “flat text” files - so you can read them using any old text editor (e.g. Notepad).
- the most commonly used document preparation system in quantitative political science, and familiarity with L^AT_EX will be expected of you.

1.2 Where can I find L^AT_EX?

To use L^AT_EX on a machine in the Star Lab, all you will need to do is to open a text editor of your choice. You can use any text editor you feel comfortable with, and there are dozens of text editors that interface with L^AT_EX to choose from.¹ Text editors differ on a number of dimensions, including: whether or not they are free, whether you see and edit the source code or the formatted text, and whether or not an in-built spell-checker or PDF previewer is included. In general, all of these text editors are graphical user interfaces that call on particular commands with the click of a button. Your colleagues will vary in which text editor they prefer, and your own preferences may come to differ from my own. In this course, we will use TeXMaker. TeXMaker is a cross-platform

¹Here is a handy comparison of the different text editors out there that come with built-in L^AT_EX functionality.

software with many of the same advantages as WinEdt, but unlike WinEdt, TeXMaker is free and has come with an in-built previewer.

However, if you are using L^AT_EX on your own machine, you will first need to install a package manager and compiler. The Star Lab machines are installed with MiKTeX (pronounced “mick-teck”), developed for the Windows OS. MiKTeX is available for download here. Alternatively, if you are using a Mac, you can use either MacTeX or XeTeX.

To view your final output, you will also need a PDF viewer like Acrobat, Foxit, Skim or Preview. On the Star Lab machines, Adobe programs are installed and available to view and edit .pdf files. (You probably already have Acrobat Reader at home.) It is worth installing a compiler and PDF viewer before installing your text editor, as most of them will query the file system to verify if and where all the necessary software has been installed.

1.3 Starter Exercise

1. Download TeXMaker onto your machine, if it is not already installed.
2. Go to <http://chitrlekha-basu.com/teaching/latex-short-course/>.
3. Save **Sample LaTeX File** to your desktop and open this file in TeXMaker.
4. Compile the document by selecting “Quick Build” on the tool bar near the top of the screen and clicking the first blue arrow from the left. View the output, either by clicking on the .pdf file that is generated on your desktop or by clicking on “Pdf Viewer” in the bottom right hand corner of your screen (if not already selected).
5. Look at your desktop. Next time, save your .tex file in a directory!

2 Basic Content in L^AT_EX

When writing a `.tex` file, you will only ever use the following characters:

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

! " # \$ % & ' () * + , - . / :

; < = > ? @ [\] ^ _ { | } ~

2.1 Characters & Commands

- L^AT_EX has ten special characters: # \$ % & \ ^ _ { } ~
- The character % tells L^AT_EX that everything after it on the same line is a comment. This means that everything between the character % and the next end-of-line character will not appear in your final output, as L^AT_EX knows to ignore it. Try typing %Hello? into your `.tex` file. What do you see?
- The backslash \ is used to begin every L^AT_EX command. You've already used a couple of L^AT_EX commands: `\noindent` tells L^AT_EX not to indent the line of the paragraph, and `\TH` (in combination with some other stuff we will get to) tells L^AT_EX to write a funny Icelandic letter and not just TH.
- The characters { and } are used to define arguments to L^AT_EX commands. If they are not matched, L^AT_EX will produce an error. TeXMaker will help you by highlighting the partner of each bracket when your cursor is next to it.
- Not all commands require arguments, but some commands are meaningless without an argument. Try typing `\begin` into TeXMaker and compile your document. This produces the following error message:

```
! Paragraph ended before \begin was complete.
<to be read again>
\par
```

1.16

I suspect you've forgotten a '}', causing me to apply this control sequence to too much text. How can we recover?

My plan is to forget the whole thing and hope for the best.

More probably, TeXMaker will prevent you from making such a mistake in the first place, by prompting you to issue an argument when necessary.

- A command in L^AT_EX may also accept optional arguments. The optional arguments to a command are passed inside square brackets instead of curly braces, like this:

```
\documentclass[12pt]{article}
```

In this case, the optional argument specifies the font size to be used. Try deleting the optional argument from this command. Does your document still compile?

- Commands are case-sensitive. What happens if you ask L^AT_EX to create `\Noindent`?
- If you want to write the symbols #, %, &, \$, { and }—in text—you will need to use a backslash before the symbol. So to write \$, you will need to enter `\$`.
- We will go over the use of (most of) the remaining special characters in due course.

2.2 Whitespace and Spacing

- Two or more carriage returns, `\\`, and `\linebreak` will move any succeeding text to a new line. A single carriage return will simply produce a non-breaking space.
- The special character `~` also produces a non-breaking space.
- The command `\par` will create a new paragraph for the succeeding text.
- The commands `\linebreak` and `\pagebreak` do as they say.
- Any number of consecutive spaces are treated by L^AT_EX as a single space.

2.3 Structure of an Input File

A $\text{\LaTeX}$ document has two main components: the Preamble and the Body. The preamble sets up the document by specifying the document class, calls on any number of packages you *might* use, and allows you to define new commands and configure properties of the document (like author, title, margins, font size, etc...). With a few exceptions (e.g., title commands) the text included in the preamble will not show up in the compiled document.

- In the document you have been working with, the preamble looks like this:

```
\documentclass[12pt]{article}

\usepackage[T1]{fontenc}
\usepackage[icelandic, english]{babel}

\title{Untitled}
\author{}
\date{\today}
```

The first command here is `\documentclass[12pt]{article}`. This tells $\text{\LaTeX}$ to use the `article` document class. We will discuss other classes of documents later in the course.

- The remainder of the preamble consists of several ‘declarations’. We tell $\text{\LaTeX}$ to load two packages which allow it to include characters from the Icelandic alphabet as well as the English alphabet, and also what the title of the article will be.² The command `\date{\today}` tells $\text{\LaTeX}$ to date the article with today’s date. If you recompile this `.tex` file at a later date, it will automatically update the date to the newer one. The `\author{}` command is left blank, and therefore does not produce any visible output in your document. Finally, the actual title and date are produced by the `\maketitle` command which occurs in the body of your document and not in the preamble.

The body of your document is enclosed within the `document` environment. $\text{\LaTeX}$ is a language that treats text differently depending on its environment, and each environment defines how the text it contains should be treated.

²You will frequently load packages to allow $\text{\LaTeX}$ to use features that are not installed by default - like writing in Icelandic...

- All of the content that will be displayed in your final document is contained in one or more environments, each bracketed by the `\begin{envt}` and `\end{envt}` commands. If you begin an environment, it must be ended.
- Other examples of environments include the `itemize` environment—which creates unordered lists—and the `enumerate` environment—which creates ordered lists. The `table` and `figure` environments are used to enclose tables and graphics respectively.
- Environments may be nested within each other, with the caveat that when nesting environments, the most recently opened environment must be closed before an earlier environment can be closed. The `document` environment is an exception: it can only occur once in a document, and is necessary if you want to write anything. (Try deleting the `\begin{document}` and `\end{document}` commands from your `.tex` file. You will not be able to compile your document.)
- Any text outside the `document` environment in your `article` will not be included in the compiled document.

3 Full Documents in L^AT_EX

3.1 Review Exercise I

1. Open a new `.tex` file in TeXMaker and save it to a directory of your choice.
2. Set the document font size to 11.
3. In the body of your document, include the following text:

```
\begin{quotation}
```

```
Four score and seven years ago our fathers brought forth on this  
continent a new nation, conceived in liberty, and dedicated to the  
proposition that all men are created equal. \\
```

```
\noindent Now we are engaged in a great civil war, testing whether  
that nation, or any nation so conceived and so dedicated, can long  
endure. We are met on a great battlefield of that war. We have come  
to dedicate a portion of that field, as a final resting place for  
those who here gave their lives that that nation might live. It is  
altogether fitting and proper that we should do this.
```

```
\end{quotation}
```

4. Compile your document. If there are errors, check that all the environments you have opened are also closed, and that all your commands are spelled correctly. If you're still stuck, raise your hand!

3.2 Typesetting Text

All text in the body of your document will be in one of two modes: paragraph or math. Paragraph mode is for ordinary text, which is all you have encountered so far. To begin a new paragraph, leave a blank line in the input file. To write in math mode, we have one of two options. To display math in-line, we use dollar signs: `$y=mx+c$` gives you $y = mx + c$. To display math on its own below your text, bracket your equations with `\[` and `\]`. So, `\[y=mx+c\]` produces

$$y = mx + c$$

We will spend a lot of time discussing how to write text in math mode shortly, but first, we will briefly delve into ways of formatting plain text in L^AT_EX.

Margins & Line Spacing

By default, L^AT_EX leaves a lot of white space as margins in your document. You can change this by loading the `any size` package and including the command `\margin size{l}{r}{t}{b}` in your preamble, where `l`, `r`, `t` and `b` is the amount of space (in either inches or centimeters) you want in your left, right, top and bottom margins, respectively. You will need to specify units for your desired margins or L^AT_EX will yell at you.

- Look back at the document we just created.
- Load the `any size` package, and include the command

```
\margin size{1 cm}{1 cm}{1 cm}{1 cm}
```

in your preamble.

- Compile your document again.
- Now, change the margins from 1 cm to 1 inch each, and compile again.

By default, your document will be single spaced. You can also change this by loading the `set space` package, and then including either the `\double spacing` or `\one half spacing` commands in your preamble.

It is worth mentioning that the default font size for the `article` class of documents is 10 pt. You can change this to either 11 pt or 12 pt by altering the optional argument to `\document class{article}` in your preamble.³

Font Styles and Sizes

To change either your font style or font size for portions of your document, you can use one of two approaches: the *environment* approach or the *command* approach. By the first approach, the text contained within the environment will be in a different font style or size. By the second approach, only the text included as the mandatory argument to the command will be altered.

³To use other default font sizes for your document, you can load the `ext sizes` package. This allows for the following font sizes: 8pt, 9pt, 10pt, 11pt, 12pt, 14pt, 17pt, 20pt. You will then need to change the document class names from e.g. `article` to `ext article`, and so forth.

Examples of font styles include **bold face** or SMALL CAPS. Some of the most common font styles in use and associated environments/commands are presented below:

Style	Environment	Command
normal/default	<code>\begin{textnormal} ... \end{textnormal}</code>	<code>\textnormal{}</code>
bold	<code>\begin{bf} ... \end{bf}</code>	<code>\textbf{}</code>
<i>italics</i>	<code>\begin{it} ... \end{it}</code>	<code>\textit{}</code>
SMALLCAPS	<code>\begin{sc} ... \end{sc}</code>	<code>\textsc{}</code>
<u>underline</u>	-NA-	<code>\underline{}</code>
Roman	<code>\begin{textrm} ... \end{textrm}</code>	<code>\textrm{}</code>
Sans Serif	<code>\begin{textsf} ... \end{textsf}</code>	<code>\textsf{}</code>
teletype	<code>\begin{tt} ... \end{tt}</code>	<code>\texttt{}</code>

In a similar vein, there are ten declarations available in L^AT_EX for altering font size *within* your document (as compared to the default font size for your document). These are presented below:

Style	Environment	Command
tiny	<code>\begin{tiny} ... \end{tiny}</code>	<code>\tiny{}</code>
scriptsize	<code>\begin{scriptsize} ... \end{scriptsize}</code>	<code>\scriptsize{}</code>
footnotesize	<code>\begin{footnotesize} ... \end{footnotesize}</code>	<code>\footnotesize{}</code>
small	<code>\begin{small} ... \end{small}</code>	<code>\small{}</code>
normalsize	<code>\begin{normalsize} ... \end{normalsize}</code>	<code>\normalsize{}</code>
large	<code>\begin{large} ... \end{large}</code>	<code>\large{}</code>
Large	<code>\begin{Large} ... \end{Large}</code>	<code>\Large{}</code>
LARGE	<code>\begin{LARGE} ... \end{LARGE}</code>	<code>\LARGE{}</code>
huge	<code>\begin{huge} ... \end{huge}</code>	<code>\huge{}</code>
Huge	<code>\begin{Huge} ... \end{Huge}</code>	<code>\Huge{}</code>

To explore how this works: create a new paragraph in your document, containing the following lines.

But, in a larger sense, we can not dedicate, we can not consecrate, we can not hallow this ground. The brave men, living and dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember what we say here, but it can never forget what they did here.

Move all your text out of the `quotation` environment, and try altering the font size and style of this paragraph to something apart from normal and something larger than `normalsize`. This will require you to “nest” the two environments.

Sections, Subsections & Subsubsections

Each document class comes with an in-built hierarchy and set of sub-headings which it accepts. An `article` may contain the following sub-headings, and in the following order: `\part{}`, `\section{}`, `\subsection{}`, `\subsubsection{}`, `\paragraph{}`, `\subparagraph{}`. For instance, we are presently in sub-sub-section 3 of sub-section 2 of section 3 of the course notes.

If an “*” is placed after the subheading name—as in `\part*{}`—then the heading will not be numbered. Alternatively, you can include the following command in your preamble:

```
\setcounter{secnumdepth}{--}
```

where the second argument to this command specifies the number of levels of sub-headings you would like numbered. You will notice that only two levels of sub-headings are numbered in this document - that is, sub-sub-sections are bolded but not numbered. This was obtained by including the command `\setcounter{secnumdepth}{2}` in the preamble to this document.

Place all of the text you have written so far into a section titled “Gettysburg Address”, and divide the text into two sub-sections. Title them whatever you want.

List Environments

There are three main list environments you will likely use: `itemize`, `enumerate` and `description`.

itemize: The `itemize` environment allows you to make un-ordered lists, each item on the list preceded by a bullet point. By default, an itemized list looks like:

- this.

However, you can change the itemized symbol by adding the symbol you want as an optional argument to the `\item` command. For example, `item[$\circ$]` produces:

- an entry that looks like this.

enumerate: The `enumerate` environment allows you to make ordered lists (by default, numbered lists). By installing the additional package `enumerate`, you can produce ordered lists with letters or Roman numerals as counters. This requires adding the optional argument `[a.]` or `[i.]` to your command to begin an `enumerate` environment. So, the command `\begin{enumerate}[a.]`⁴ produces

a. something like this.

description: The `description` environment is slightly different from the other list environments we have discussed, in that the user must specify the “term” being described in each entry. So, when calling upon the `description` environment, we write code like the following:

```
\begin{description}
\item[An item] is blue.
\item[Another item] is red.
\item[A third item] is purple.
\end{description}
```

To create the list you are reading right now, I am using the `description` environment.

You can nest these list environments within each other. The nested environments do not need to be of the same type: you can, for instance, nest an `enumerate` environment within an `itemize` environment. So the following code:⁵

```
\begin{itemize}

\item Abraham Lincoln, who delivered the Gettysburg Address, is one of
      43 former presidents of the United States. Other former presidents
      include:

\begin{enumerate}

\item James Madison
```

⁴As always, you will have to close every environment you begin!

⁵You will notice that I have been quite liberal with white space in the code below. This is a useful way of exploiting L^AT_EX’s conservatism in interpreting white space as space when organizing your code.

```
\item Andrew Jackson
```

```
\item Woodrow Wilson
```

```
\end{enumerate}
```

```
\item Abraham Lincoln was portrayed by Daniel Day-Lewis in the 2012 film  
      directed by Steven Spielberg. Daniel Day-Lewis grew a beard for the  
      role.
```

```
\end{itemize}
```

produces:

- Abraham Lincoln, who delivered the Gettysburg Address, is one of 43 former presidents of the United States. Other former presidents include:
 1. James Madison
 2. Andrew Jackson
 3. Woodrow Wilson
- Abraham Lincoln was portrayed by Daniel Day-Lewis in the 2012 film directed by Steven Spielberg. Daniel Day-Lewis grew a beard for the role.

Finally, you will have noticed that TeXMaker will prompt you with the command it thinks you want as you begin to call on a particular environment. This applies to most commands that you will use, whether to call on a particular font style or document class. You can accept its suggestion by clicking enter before you have finished typing your command, with your desired command selected from the drop-down menu that pops up. This may save you some time!

Punctuation

Finally, we deal with a few of the finer (but minor) points of typesetting text.

You will likely need, at some point in your career, to use quotation or speech marks. If you use speech marks as you might in a typical word processor, as when writing "And it was all a dream", you will find that you have written "And it was all a

dream”. To avoid this, you should write ‘‘And it was all a dream.’’—which produces: “And it was all a dream.” The same applies with quotation marks.

There are three types of dashes in L^AT_EX.

1. The **triple dash**, or three consecutive hyphens, is used when you want the reader to read the text between the dashes as if it was in brackets. Therefore,

```
Tony Kushner---already famous for his award-winning play \textit{Angels
in America}---was hired as the screenwriter for Steven Spielberg's
\textit{Lincoln}
```

yields: Tony Kushner—already famous for his award-winning play *Angels in America*—was hired as the screenwriter for Steven Spielberg’s *Lincoln*.

2. The **double dash** is made with two consecutive hyphens, and used when you want to join names that are linked. For example:

```
the Gibbard--Satterwaite theorem is on pages 56--67
```

yields: the Gibbard–Satterwaite theorem is on pages 56–67.

Finally, footnotes come in handy.⁶ To create a footnote, use the `\footnote{}` command, where the text you want to relegate to a footnote goes in the braces. Footnotes are automatically numbered. The above footnote was generated with the following code:

```
\footnote{If in doubt, note that this is footnote 6.}
```

3.3 Typesetting Math

Most of what you will do with L^AT_EX, and what it is most useful for, is typesetting even complex mathematical equations quickly and elegantly. To avoid writing a lot of additional and complicated code, it is helpful to load two particular packages every time you want to write a document with any mathematical content: `amsmath` and `amssymb`.

Open a new `.tex` file and enter the following:

⁶If in doubt, note that this is footnote 6.

```

\documentclass[12pt]{article}
\usepackage{amsmath,amssymb}

\begin{document}

\end{document}

```

We will use this document to practice mathematical commands. As you can see, it is possible to load more than one package using the same `\usepackage` command, so long as each package is separated from the next by a comma. You may also load the packages with separate commands (this is entirely a matter of preference).

Math Modes

In order to display mathematical content, L^AT_EX requires that you either be in math mode, or in a mathematical environment where the math mode is implied. Examples of such environments are `align` or `equation`. We have already briefly discussed two ways of writing in math mode: to display math in-line, we place the desired expression between two `$` symbols; however, to display math on its own below your text, we bracket our equations with `\[` and `\]`.

When comparing the two modes, the differences in how the resulting math looks on the compiled page sometimes goes beyond indentation and size. For instance, $\int_0^\infty \sum_{n=1}^\infty \frac{1}{n} dx$ and

$$\int_0^\infty \sum_{n=1}^\infty \frac{1}{n} dx$$

are generated using the same code but in different math modes. To reproduce the same equation in your `.tex` file, enter the following code into your document while in math mode:

```
\int^\infty_0\sum^\infty_{n=1}\frac{1}{n} dx
```

Note that letters in math mode look very different to those in paragraph mode. So `$dx$` gives dx , while in text, this would be rendered `dx`. While in math mode, you can force text to appear *as text* using the `\text` command. Then,

```
\int^\infty_0\sum^\infty_{n=1}\frac{1}{n} \text{dx}
```

produces

$$\int_0^\infty \sum_{n=1}^\infty \frac{1}{n} dx$$

Subscripts and Superscripts

To add a superscript to an expression, we use a hat, “^”, after the term with an exponent. So, to write X^2 , we enter `$X^2$`. However, if your exponent has more than one character to be rendered - as with X^{2+y} , then we must surround the entire exponent in curly braces (`{}`). Without the curly braces, `$X^2+y$` is rendered as $X^2 + y$. Subscripts are produced analogously, except that we use the underscore, “_”, instead.

Both super- and subscripts can be nested, but require careful use of braces to group terms in such cases. For instance, we can typeset e^{x_i} with `$e^{\texttt{x}_{i}}$`. The braces around the `i` term are optional. Remove them - what happens to the rendering? Now, remove the braces around the x_i term as well and compile the document. What happens to the rendering now?

Finally, subscripts and superscripts can be used simultaneously. For instance, in Social Choice theory, we often care about the median voter’s preference, denoted x_i^m . This can be achieved with either `\texttt{x}_m^i` or `\texttt{x}_{m}^{\texttt{i}}`. For integration and sums, the same applies. To indicate integration between 0 and ∞ —as in the expression above—we write `$\int^{\infty}_0$`; to sum between $n = 1$ and ∞ over $\frac{1}{n}$, we write `\sum^{\infty}_{n=1} \frac{1}{n}`.

Operators, etc.

You will find (essentially) every mathematical symbol you will ever need in TeXMaker’s menu of symbols. Click on the ‘Structure’ tab in the bottom left corner of your screen, which will bring up a whole list of new buttons you can click. Try clicking on the first few buttons from the top left. You will find yourself confronted with a smorgasbord of symbols to choose from. For the more persistent among you...

$+$	$\vee$	$\forall$
$=$	$\wedge$	$\exists$
$-$	$\bigvee$	$\subset$
$/$	$\bigwedge$	$\subseteq$
$<$	$\cup$	$\subsetneq$
$>$	$\cap$	$\supset$
$\leq$	$\bigcup$	$\supseteq$
$\geq$	$\bigcap$	$\supsetneq$
$\int$	$\Leftrightarrow$	$\neg$
$\sum$	$\leftrightarrow$	$\neg$
$\prod$	$\Leftarrow$	$\div$
∂	$\leftarrow$	∞
$\sim$	$\Rightarrow$	$\emptyset$
$\approx$	$\rightarrow$	$\in$

Greek Letters

You will soon know more Greek than you ever thought you would. Although there are no ω or ψ keys on your keyboard, L^AT_EX allows you to include these symbols while in math mode in a very intuitive way. You can type `\name-of-letter` or `\Name-of-letter` for the lower-case and upper-case version of each Greek letter. So, typing `\theta` gives you θ , while typing `\Theta` gives you Θ . However, not all Greek letters have an upper- and lower-case version. For example, although `\beta` gives β , the command `\Beta` is undefined.

If you're anything like me, you probably can't remember what ξ is called. Again, TeX-Maker is here to help you. Click on λ to find a sizeable menu of Greek letters to choose from. Clicking on any of these letters will print the appropriate command in your document; note that you will need to ensure the command is in math mode or L^AT_EX will not know what to do.

Finally, if you write a Greek letter command followed by an operator—like $+$, $-$, $\times$, or $\div$ —you **must** include a space between the two. The command `\alpha + \epsilon` is interpreted by L^AT_EX as $\alpha + \epsilon$, but `\alpha+\epsilon` will produce an error, as L^AT_EX will assume you meant to write `\alpha+`, an undefined command.

Accents

Suppose you want to indicate the estimate $\hat{\beta}$ and not the true parameter β . This will require the use of accents: to obtain $\hat{\beta}$, we write `\hat{\beta}`. Note that you will need to be in math mode for this to work. Here are a few of the more commonly required accents.

$\acute{a}$ <code>\acute{a}</code>	$\dot{e}$ <code>\dot{e}</code>	$\tilde{i}$ <code>\tilde{\imath}</code>
$\bar{b}$ <code>\bar{b}</code>	$\ddot{f}$ <code>\ddot{f}</code>	$\vec{j}$ <code>\vec{j}</code>
$\breve{c}$ <code>\breve{c}</code>	$\grave{g}$ <code>\grave{g}</code>	$\widehat{xyz}$ <code>\widehat{xyz}</code>
$\check{d}$ <code>\check{d}</code>	$\hat{h}$ <code>\hat{h}</code>	$\widetilde{xyz}$ <code>\widetilde{xyz}</code>

Words: `\lim`, `\det`, `\sin`...

When writing mathematical expressions, we often use abbreviated words - `\cos` to mean cosine, `\ln` to indicate the natural logarithm, and so on. If we were to just write these words as text in math mode, they would not be rendered properly. For example: `\cos\theta` produces $\cos\theta$. However, `\cos\theta` instead produces $\cos\theta$. The commands associated with frequently used expressions are given below (there are no real surprises here):

$\acute{a}$ <code>\acute{a}</code>	$\dot{e}$ <code>\dot{e}</code>	$\tilde{i}$ <code>\tilde{\imath}</code>
$\bar{b}$ <code>\bar{b}</code>	$\ddot{f}$ <code>\ddot{f}</code>	$\vec{j}$ <code>\vec{j}</code>
$\breve{c}$ <code>\breve{c}</code>	$\grave{g}$ <code>\grave{g}</code>	$\widehat{xyz}$ <code>\widehat{xyz}</code>
$\check{d}$ <code>\check{d}</code>	$\hat{h}$ <code>\hat{h}</code>	$\widetilde{xyz}$ <code>\widetilde{xyz}</code>

Fractions

You have already encountered a few fractions in the expressions we have used. There are several ways of writing fractions in L^AT_EX. The easiest approach is to write `x/y` which yields x/y . However, especially when dealing with complex fractions, your final output will be easier to read and interpret if you use the `\frac{}{}` command. Generally, this command takes the following form:

`\frac{numerator}{denominator}`

so `\frac{dy}{dx}` yields:

$$\frac{dy}{dx}$$

You can use fractions in subscripts or superscripts. So, `e^{\frac{1}{x}}` yields

$$e^{\frac{1}{x}}$$

With a similar command, you can write binomial-like expressions too: `\binom{n}{i}` yields

$$\binom{n}{i}$$

Brackets & Delimiters

Brackets are useful. The expression $(100 - 100)^{100}$ means something quite different from the expression $100 - 100^{100}$. For simple expressions, the commands are just as you would expect. For parentheses, we use `(x)`. For curly braces, we use `{x}`. Last but not least, for square brackets, we use `[x]`. These produce (x) , x , and $[x]$, respectively.

However, for longer expressions, these commands can lead to funny results. For example,

$$(\sum_{i=1}^N x_i)$$

looks funny. To fix this, we instead write:

`\left(\sum_{i=1}^N x_i\right)`

which produces

$$\left(\sum_{i=1}^N x_i\right)$$

The analogous commands for curly braces and square brackets are as follows: `\left\{`, `\right\}`, `\left[` and `\right]`. Finally, note that the commands necessary to create brackets and delimiters are the same in both paragraph and math mode.

Math Mode Miscellany

To get extra space between characters on a line in paragraph mode, we can use the `\hspace{}` command. This command does nothing within math mode. So to produce extra space between numbers and/or operators in math mode, we use the command `\;`. This generates a teensy bit of extra space. To produce a noticeable difference, you will need to use this command multiple times in succession. For instance, `x+\cos\;\;\;\;\;\theta` produces

$$x + \cos \quad \theta$$

Modification	Command	Example	Common Use
Normal Math Face	—	ABCXY	most maths
Roman Face	<code>\textrm{}</code>	ABCXY	text within equation
Bold Math Face	<code>\mathbf{}</code>	ABCXY	vectors and matrices
Blackboard Math	<code>\mathbb{}</code>	$\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{X}\mathbb{Y}$	special sets of numbers
Calligraphic Math	<code>\mathcal{}</code>	$\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{X}\mathcal{Y}$	arbitrary sets

You may need to change the presentation (or “typeface”) of letters or symbols while in math mode. The most commonly required modifications and the associated commands are listed above.

3.4 Common Errors and Debugging

There will always be times when $\text{\LaTeX}$ won’t do what you’re trying to tell it to do, and will instead respond with something like this:

```
! Paragraph ended before \begin was complete.<to be read again>\par
```

Although it may be tempting, the best thing to do in such times is not to hurl your computer at a wall. First, note that TeXMaker will try and help you identify what went wrong. In this case, an environment was started but not closed. It will also tell you which line of code was at fault. By clicking on the error message, you will immediately be brought to the offending code. Compile your document again when you think you have fixed the problem.

The usual suspects: incorrectly entering a command, missing a closing brace, using a command in the wrong mode, or missing a backslash before a “special character”. This will get easier with time!

A Appendix

For future reference, here's a cheat sheet.⁷

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	τ	<code>\tau</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	γ	<code>\gamma</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	φ	<code>\varphi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	χ	<code>\chi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>	ω	<code>\omega</code>
η	<code>\eta</code>	ξ	<code>\xi</code>				
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

Table 1: Greek Letters

$\pm$	<code>\pm</code>	$\cap$	<code>\cap</code>	$\diamond$	<code>\diamond</code>	$\oplus$	<code>\oplus</code>
$\mp$	<code>\mp</code>	$\cup$	<code>\cup</code>	$\triangleup$	<code>\triangleup</code>	$\ominus$	<code>\ominus</code>
$\times$	<code>\times</code>	$\uplus$	<code>\uplus</code>	$\triangledown$	<code>\triangledown</code>	$\otimes$	<code>\otimes</code>
$\div$	<code>\div</code>	$\sqcap$	<code>\sqcap</code>	$\triangleleft$	<code>\triangleleft</code>	$\oslash$	<code>\oslash</code>
$*$	<code>\ast</code>	$\sqcup$	<code>\sqcup</code>	$\triangleright$	<code>\triangleright</code>	$\odot$	<code>\odot</code>
$\star$	<code>\star</code>	$\vee$	<code>\vee</code>	$\triangleleft^b$	<code>\triangleleft^b</code>	$\bigcirc$	<code>\bigcirc</code>
$\circ$	<code>\circ</code>	$\wedge$	<code>\wedge</code>	$\triangleright^b$	<code>\triangleright^b</code>	$\dagger$	<code>\dagger</code>
$\bullet$	<code>\bullet</code>	$\setminus$	<code>\setminus</code>	$\unlhd^b$	<code>\unlhd^b</code>	$\ddagger$	<code>\ddagger</code>
$\cdot$	<code>\cdot</code>	$\wr$	<code>\wr</code>	$\unrhd^b$	<code>\unrhd^b</code>	$\amalg$	<code>\amalg</code>
$+$	<code>+</code>	$-$	<code>-</code>				

⁷The following figures were prepared by L. Kocbach and based on a document originally prepared by David Carlisle (University of Manchester). You can find the original source material [here](#).

$\leq$	<code>\leq</code>	$\geq$	<code>\geq</code>	$\equiv$	<code>\equiv</code>	$\models$	<code>\models</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>	$\perp$	<code>\perp</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>	$\mid$	<code>\mid</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\asymp$	<code>\asymp</code>	$\parallel$	<code>\parallel</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>	$\bowtie$	<code>\bowtie</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>	$\Join^b$	<code>\Join^b</code>
$\sqsubset^b$	<code>\sqsubset^b</code>	$\sqsupset^b$	<code>\sqsupset^b</code>	$\neq$	<code>\neq</code>	$\smile$	<code>\smile</code>
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\doteq$	<code>\doteq</code>	$\frown$	<code>\frown</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni</code>	$\propto$	<code>\propto</code>	$=$	<code>=</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$<$	<code><</code>	$>$	<code>></code>
$:$	<code>:</code>						

`,` `,` `;` `;` `:` `\colon` `.` `\ldotp` `.` `\cdotp`

Table 4: Punctuation Symbols

$\leftarrow$	<code>\leftarrow</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\rightarrow$	<code>\rightarrow</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Longleftrightarrow$	<code>\Longleftrightarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>	$\nearrow$	<code>\nearrow</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>	$\searrow$	<code>\searrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\swarrow$	<code>\swarrow</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightharpoonleft$	<code>\rightharpoonleft</code>	$\leadsto^b$	<code>\leadsto^b</code>		

$\dots$	<code>\ldots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>
$\aleph$	<code>\aleph</code>	$\prime$	<code>\prime</code>	$\forall$	<code>\forall</code>	∞	<code>\infty</code>
$\hbar$	<code>\hbar</code>	$\emptyset$	<code>\emptyset</code>	$\exists$	<code>\exists</code>	$\Box^b$	<code>\Box^b</code>
$\imath$	<code>\imath</code>	∇	<code>\nabla</code>	$\neg$	<code>\neg</code>	$\Diamond^b$	<code>\Diamond^b</code>
$\jmath$	<code>\jmath</code>	$\surd$	<code>\surd</code>	$\flat$	<code>\flat</code>	$\triangle$	<code>\triangle</code>
ℓ	<code>\ell</code>	$\top$	<code>\top</code>	$\natural$	<code>\natural</code>	$\clubsuit$	<code>\clubsuit</code>
$\wp$	<code>\wp</code>	$\bot$	<code>\bot</code>	$\sharp$	<code>\sharp</code>	$\diamondsuit$	<code>\diamondsuit</code>
$\Re$	<code>\Re</code>	$\parallel$	<code>\parallel</code>	$\backslash$	<code>\backslash</code>	$\heartsuit$	<code>\heartsuit</code>
$\Im$	<code>\Im</code>	$\angle$	<code>\angle</code>	∂	<code>\partial</code>	$\spadesuit$	<code>\spadesuit</code>
$\mho^b$	<code>\mho^b</code>	$.$	<code>.</code>	$ $	<code> </code>		

$\sum$	<code>\sum</code>	$\bigcap$	<code>\bigcap</code>	$\bigodot$	<code>\bigodot</code>
$\prod$	<code>\prod</code>	$\bigcup$	<code>\bigcup</code>	$\bigotimes$	<code>\bigotimes</code>
$\coprod$	<code>\coprod</code>	$\bigsqcup$	<code>\bigsqcup</code>	$\bigoplus$	<code>\bigoplus</code>
$\int$	<code>\int</code>	$\bigvee$	<code>\bigvee</code>	$\biguplus$	<code>\biguplus</code>
$\oint$	<code>\oint</code>	$\bigwedge$	<code>\bigwedge</code>		

Table 7: Variable-sized Symbols

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>	<code>\min</code>	<code>\sinh</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>	<code>\Pr</code>	<code>\sup</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>	<code>\sec</code>	<code>\tan</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\sin</code>	<code>\tanh</code>

Table 8: Log-like Symbols

<code>(</code>	<code>(</code>	<code>)</code>	<code>)</code>	$\uparrow$	<code>\uparrow</code>	$\Uparrow$	<code>\Uparrow</code>
<code>[</code>	<code>[</code>	<code>]</code>	<code>]</code>	$\downarrow$	<code>\downarrow</code>	$\Downarrow$	<code>\Downarrow</code>
<code>{</code>	<code>\{</code>	<code>}</code>	<code>\}</code>	$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
<code>\lfloor</code>	<code>\lfloor</code>	<code>\rfloor</code>	<code>\rfloor</code>	$\lceil$	<code>\lceil</code>	$\rceil$	<code>\rceil</code>
<code>\langle</code>	<code>\langle</code>	<code>\rangle</code>	<code>\rangle</code>	<code>/</code>	<code>/</code>	<code>\</code>	<code>\backslash</code>
<code> </code>	<code> </code>	<code>\ </code>	<code>\ </code>				

Table 9: Delimiters

<code>\rmoustache</code>	<code>\lrmoustache</code>	<code>\rgroup</code>	<code>\lgroup</code>
<code>\arrowvert</code>	<code>\Arrowvert</code>	<code>\bracevert</code>	

Table 10: Large Delimiters

$\hat{a}$	<code>\hat{a}</code>	$\acute{a}$	<code>\acute{a}</code>	$\bar{a}$	<code>\bar{a}</code>	$\dot{a}$	<code>\dot{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\check{a}$	<code>\check{a}</code>	$\grave{a}$	<code>\grave{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>

Table 11: Math mode accents

$\widetilde{abc}$	<code>\widetilde{abc}</code>	$\widehat{abc}$	<code>\widehat{abc}</code>
$\overleftarrow{abc}$	<code>\overleftarrow{abc}</code>	$\overrightarrow{abc}$	<code>\overrightarrow{abc}</code>
$\overline{abc}$	<code>\overline{abc}</code>	$\underline{abc}$	<code>\underline{abc}</code>
$\overbrace{abc}$	<code>\overbrace{abc}</code>	$\underbrace{abc}$	<code>\underbrace{abc}</code>
$\sqrt{abc}$	<code>\sqrt{abc}</code>	$\sqrt[n]{abc}$	<code>\sqrt[n]{abc}</code>
f'	<code>f'</code>	$\frac{abc}{xyz}$	<code>\frac{abc}{xyz}</code>

Table 12: Some other constructions