

Hendrix L^AT_EX checklist for students

January 17, 2023

L^AT_EX is the *de facto* standard system for creating and distributing high-quality documents in mathematics and computer science, and you may be required to use it in several classes throughout the Mathematics and Computer Science majors.

For creating L^AT_EX documents, we recommend using Overleaf, a free online editor that even allows multiple people to easily collaborate on the same document. For learning about the basics of L^AT_EX, we recommend working through some of the exercises in the Bates LaTeX Manual, or the Overleaf tutorials.

This document is a simple checklist you can use to ensure you are following some basic best practices for creating professional-looking L^AT_EX documents. It can be found at <https://github.com/Hendrix-CS/LaTeX-checklist/blob/main/LaTeX-checklist.tex>.

Throughout this document, examples of **bad** L^AT_EX style are typeset in a red box, like the one below. You can see the L^AT_EX code in the top half of the box, and the resulting typeset document on the bottom.

`$2^{120}` is a big number\$. Don't copy the red boxes!

`2^{120}` is a big number. Don't copy the red boxes!

Examples of **good** style, on the other hand, are typeset in a green box, like this:

`$2^{\{120\}}` is a big number. Pay attention to the green boxes!

2^{120} is a big number. Pay attention to the green boxes!

Formatting

- Use a blank line as separation between paragraphs.

This is the first paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

This is another paragraph. Normally the start of the paragraph will be indented for you automatically, though it looks a bit different inside this green box.

This is the first paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

This is another paragraph. Normally the start of the paragraph will be indented for you automatically, though it looks a bit different inside this green box.

- Use `` (two back ticks) and '' (two single quotes) to create quotation marks:

``like this!''

“like this!”

If you use normal double quote characters, the result will be typeset using two “close quote” characters.

"These quote marks look bad."

”These quote marks look bad.”

- Use `\emph{...}` to italicize things like Latin abbreviations (such as *e.g.* and *i.e.*) or words you are going to define.

A `\emph{snoozle}` is a special kind of matrix which has all positive entries, `\emph{i.e.}` all the entries are greater than zero.

A *snoozle* is a special kind of matrix which has all positive entries, *i.e.* all the entries are greater than zero.

- There are four kinds of horizontal lines:

- A minus sign is created by using a hyphen inside math mode.

`$(-2) - 3$`

$(-2) - 3$

`x = -4 % A minus sign outside of math mode`

$x = -4$

- A single hyphen is used for hyphenated words.

`helter-skelter`

helter-skelter

- A double hyphen creates an “en-dash”, which is used for ranges.

`Pages 23--69 discuss the period 1952--1987.`

Pages 23–69 discuss the period 1952–1987.

- A triple hyphen creates an “em-dash”, which is used as punctuation to signal a break in thought. The usual typesetting style (in the US, at least) is to leave no spaces on either side of the em-dash.

`If this were true---which it isn't---then`

If this were true— which it isn’t — then

- Use `itemize` or `enumerate` for lists.

```

\begin{itemize}
  \item One thing
  \item Another thing
    \begin{itemize}
      \item Lists can
      \item be nested
    \end{itemize}
  \item Last thing
\end{itemize}

```

-
- One thing
 - Another thing
 - * Lists can
 - * be nested
 - Last thing

Using `enumerate` instead of `itemize` causes the items to be sequentially numbered.

```

\begin{enumerate}
  \item One thing
  \item Another thing
  \item Last thing
\end{enumerate}

```

-
1. One thing
 2. Another thing
 3. Last thing

Don't just try to make your own bulleted list manually.

```

* This looks bad

* And also won't have the right amount of space around it

```

```

* This looks bad
* And also won't have the right amount of space around it

```

- If you use a command in the middle of some text, by default it will eat any following whitespace:

```
\LaTeX is great
```

```
LATEXis great
```

Follow the command with a backslash before the space to force an explicit space to appear, like this:

```
\LaTeX\ is great
```

```
LATEX is great
```

- \LaTeX automatically puts extra space at the end of every sentence, but it thinks any punctuation is the end of a sentence, so abbreviations can have too much space after them. Use a backslash before the space to force a normal space. (But this is kind of nitpicky since it can be hard to even tell the difference!)

```
Dr. Yorgey likes \LaTeX
```

```
Dr. Yorgey likes LATEX
```

```
Dr.\ Yorgey likes \LaTeX
```

```
Dr. Yorgey likes LATEX
```

Math

- Use the `amsmath` and `amssymb` packages by putting `\usepackage{amsmath}` and `\usepackage{amssymb}` in the *preamble* of your document (that is, the part before `\begin{document}`).
- Use math mode appropriately:
 - Use dollar signs for inline mathematics (including variable names, numbers, equations, formulas, ...)

In this case, x cannot possibly be -3 , because we assumed that $x > 10$.

In this case, x cannot possibly be -3 , because we assumed that $x > 10$.

- Use `\[... \]` for formulas or equations that should be centered on their own line.

Solving for x , we find that `\[x = 2 + y^2 - \pi, \]` which is what we wanted to show.

Solving for x , we find that

$$x = 2 + y^2 - \pi,$$

which is what we wanted to show.

- Use `$... $` or `\[... \]` for an entire equation/formula, instead of building it out of lots of little pieces. Use `\text{...}` if you need some normal text in the middle of your formula.

`$x = \pm 2$ or $(\sqrt{7} - \pi)$`

$x = \pm 2$ or $(\sqrt{7} - \pi)$

`$x = \pm 2$ \text{ or } $(\sqrt{7} - \pi)$`

$x = \pm 2$ or $(\sqrt{7} - \pi)$

`\[f(x) = \begin{cases} x^2 & \text{if } x > 0 \\ x & \text{otherwise} \end{cases} \]`

$$f(x) = \begin{cases} x^2 & \text{if } x > 0 \\ x & \text{otherwise} \end{cases}$$

- Equations and formulas should be used as part of complete sentences with proper punctuation.

Solving for x , we find that $x = 2 + y^2 - \pi$ and therefore $x > 2z$.

Solving for x , we find that

$$x = 2 + y^2 - \pi,$$

and therefore

$$x > 2z.$$

Solving for x $x = 2 + y^2 - \pi$ $x > 2z$

Solving for x

$$x = 2 + y^2 - \pi$$

$$x > 2z$$

- Use `\mathit{...}` for longer variable names in math mode. Without this command, L^AT_EX thinks all the letters represent individual variables being multiplied together, and will insert extra space between them. For example, notice how much space is between the letters F and u below:

`$myFunction(x) \times 2^n$`

$myFunction(x) \times 2^n$

With the `\mathit{...}` command, L^AT_EX knows that the entire thing is supposed to be one word:

`$\mathit{myFunction}(x) \times 2^n$`

$\mathit{myFunction}(x) \times 2^n$

- Use commands like `\log`, `\sin`, `\cos`, `\lim` appropriately. Typically such “operator words” are typeset with upright letters, unlike variables which are typeset in italics. Using the appropriate commands ensures these operators are formatted correctly, with appropriate space around them.

`$n \log(n) + \lim_{x \to \infty} \frac{\cos \theta}{x^2}$`

$n \log(n) + \lim_{x \rightarrow \infty} \frac{\cos \theta}{x^2}$

`$n \log(n) + \lim_{x \to \infty} \frac{\cos \theta}{x^2}$`

$n \log(n) + \lim_{x \rightarrow \infty} \frac{\cos \theta}{x^2}$

- Use `\dots` for ellipses and `\cdots` for centered ellipses. Using actual periods looks too squashed:

`$\{1, 2, ..., 10\}$`

$\{1, 2, ..., 10\}$

`$1 + 2 + ... + 10$`

$1 + 2 + ... + 10$

The `\dots` and `\cdots` (centered dots) commands produce more evenly spaced dots:

`$\{1, 2, \dots, 10\}$`

$\{1, 2, \dots, 10\}$

`$1 + 2 + \cdots + 10$`

$1 + 2 + \cdots + 10$

- Use `\begin{align*}` ... `\end{align*}` for multi-step derivations.


```
\begin{align*}
f(n) &= x + x^2 + \cdots + x^n \\
&= x(1 + x + \cdots + x^{n-1}) \\
&= x \frac{x^n - 1}{x - 1}
\end{align*}
```

$$\begin{aligned}
 f(n) &= x + x^2 + \cdots + x^n \\
 &= x(1 + x + \cdots + x^{n-1}) \\
 &= x \frac{x^n - 1}{x - 1}
 \end{aligned}$$

- Use `\mathbb{...}` for appropriate symbols.

```
\mathbb{R} - \mathbb{N}
```

$$\mathbb{R} - \mathbb{N}$$

- Use `{...}` around multi-character exponents, subscripts, *etc.*

```
$2^15$
```

$$2^{15}$$

```
$2^{\{15\}}$
```

$$2^{15}$$

- Use `\left` and `\right` to make sure delimiters are big enough. In this first example, notice how the parentheses are not tall enough to enclose everything inside them:

```
$(\frac{\pi}{\sum_{k=1}^{\infty} \frac{1}{k^2}} \frac{1}{k^2})$
```

$$\left(\frac{\pi}{\sum_{k=1}^{\infty} \frac{1}{k^2}} \frac{1}{k^2}\right)$$

Putting the `\left` and `\right` commands before the parentheses ensures they are big enough:

```
$\left( \frac{\pi}{\sum_{k=1}^{\infty} \frac{1}{k^2}} \right)$
```

$$\left(\frac{\pi}{\sum_{k=1}^{\infty} \frac{1}{k^2}} \right)$$

Code

- Use the `minted` package¹ to typeset nicely syntax-highlighted code. A `verbatim` environment is not very flexible and can only produce black and white text:

```
\begin{verbatim}
def facRec(n: int) -> int:
    if n <= 1:
        return n
    else:
        return n * facRec(n-1)
\end{verbatim}

def facRec(n: int) -> int:
    if n <= 1:
        return n
    else:
        return n * facRec(n-1)
```

The `minted` package can produce code with syntax highlighting, and has many other options as well, such as line numbering, including code from an external file, *etc.*

¹https://www.overleaf.com/learn/latex/Code_Highlighting_with_minted

```

\begin{minted}{python}
def facRec(n: int) -> int:
    if n <= 1:
        return n
    else:
        return n * facRec(n-1)
\end{minted}

def facRec(n: int) -> int:
    if n <= 1:
        return n
    else:
        return n * facRec(n-1)

```

- Use a contrasting style like a typewriter font for code, names of functions, *etc.*

When we call facRec(10),

When we call facRec(10),

When we call \texttt{facRec(10)},

When we call facRec(10),

Even better, use the `mintinline` command from the `minted` package:

When we call \mintinline{python}{facRec(10)},

When we call facRec(10),

Organization

- Use `\chapter`, `\section`, `\subsection`, *etc.* appropriately to create sections within your document.
- Use `\label` and `\ref` appropriately to refer to numbered sections, theorems, figures, *etc.*