

# Evan Chen 《陳誼廷》



Teammate Hunt 2025 Author Notes 2025-04-23



[web.evanchen.cc](https://web.evanchen.cc)



## Evan's LaTeX Style Guide

This document describes the specifications requested by Evan for any LaTeX source code that is sent to him. Of course, others should feel free to use or adopt it freely.

See also the [list of pet peeves](#).

### Keywords #

The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC 2119](#).

### Requirements #

1. Source lines **MUST** be wrapped to be at most 100 characters long, except in situations where this is impossible (e.g. long URL), and **SHOULD** be wrapped to at most 80 characters long. (Note this is referring to lines in the *source code*, not the output.)

More importantly, line breaks **SHOULD** be inserted in natural places like the end of sentences, after commas, or between phrases and clauses. There is nothing wrong with short lines if it improves legibility. (One is *not* expected to use up all 80-100

- characters per line; it's more important to put line breaks in readable places than to use all available screen space.<sup>1)</sup>
2. Quotation marks MUST be `inputted correctly in LaTeX`. You MUST NOT type a literal quotation mark `"`.
  3. Operators like `\sin` MUST be typeset correctly, either using builtin commands or using the `\operatorname` command.
  4. Ellipses MUST be written with amsmath's `\dots` except in the rare cases where the auto-detection would cause the wrong type of ellipses to be used; in that case, use the appropriate `AMSMath semantic dots`, that is, one of `\dotsc`, `\dotso`, `\dotsb`, `\dotsi`, `\dotsm`. The older commands `\ldots` or `\cdots` MUST NOT be used.
  5. Grammatical commas and grammatical periods MUST NOT appear in inline math. For example:
    - `Thus $x=3.$` is not acceptable and MUST be typeset as `Thus $x=3$.` instead.
    - `$a,b,$ and $c$` is not acceptable and MUST be typeset as `$a$, $b$, and $c$` instead.
    - `Let $x_1, \dots, x_n$ be integers` is not acceptable and MUST be typeset as `Let $x_1$, $\dots$, $x_n$ be integers` instead.

Obviously, mathematical commas like those in `$f(a,b)$` and `$_{a,b,c}$` should still be in the dollar signs. This applies only to grammatical commas.
  6. There MUST NOT be extraneous spaces preceding punctuation or around dollar signs. Using two spaces after a sentence is OPTIONAL.
  7. There SHOULD generally be spaces around binary operators and relations such as `+` or `=`, but these spaces MAY be omitted for short expressions. The use of spacing MUST be symmetric, e.g. `$x =3$` is not acceptable.
  8. Delimiters of complex expressions SHOULD be balanced with `\left` and `\right`, or variants of `\big`.

9. Approximations of predefined operators MUST NOT be used. This means `||` MUST NOT be used in place of `\parallel`, or `.` in place of `\cdot`, etc.
10. For single-line display math, double dollar signs MUST NOT ever be used; the use of `\[ ... \]` instead is REQUIRED. Inserting line breaks after `\[` and before `\]` is OPTIONAL; if the line break is not included there MUST be a space instead. There SHOULD be single line break before and after displayed expressions; but there MUST NOT be double line breaks unless it is intentional that the displayed line should be its own paragraph (which is almost never the case).
11. When a series of equation are too long to fit on a single line, one MUST NOT have adjacent `\[ ... \]` expressions. In most cases, the `align*` environment SHOULD be used instead.
12. When using `align*`, the invocations `\begin{align*}` and `\end{align*}` MUST be on their own line. There MUST be a line break after each `\\` invocation. There MAY be additional line breaks for legibility, and there MUST be additional line breaks if they are necessary to keep the line length within the specified limit.
13. The contents of any `\begin{...} ... \end{...}` environment SHOULD be indented by at least two spaces.
14. There MUST NOT be any trailing whitespace, i.e. no line may end with a whitespace character.
15. Paragraph breaks MUST be typeset using two or more line breaks (i.e. one empty line between the paragraphs), and MUST NOT be typeset using `\\` or other similar antics.
16. Any mathematical variables MUST be enclosed in dollar signs, e.g. `\let n=2022` is not acceptable and MUST be typeset as `\let $n=2022$`. This also includes constants like 1 used in a mathematical context, e.g. `add 1 to both sides` is not acceptable and MUST be typeset as `add $1$ to both sides`.
17. When specifying domains and ranges of mathematical functions, one MUST use `\colon` instead of `:`, e.g. `$f \colon`

`\mathbb{R}` `\to \mathbb{R}`\$.

## Example #

The following is a solution to [AIME II 2022 Problem 13](#).

Expand the generating function and take mod  $x^{2023}$  to find that  $P(x)$  is given by

```
\begin{align*}
& (-1)^{11-4} \cdot (1+x^{30}+x^{60}+\dotsb) \\
& \cdot (1+x^{42}+x^{84}+\dotsb) \cdot \\
& \cdot (1+x^{70}+x^{140}+\dotsb) \\
& \cdot (1+x^{105}+x^{210}+\dotsb).
\end{align*}
```

So it is equivalent to find the number of quadruples of nonnegative integers  $(a,b,c,d)$  which satisfy

```
\[ 105a + 70b + 42c + 30d = 2022. \]
```

By considering modulo  $2$ ,  $3$ ,  $5$ ,  $7$  we obtain

```
\begin{align*}
a & \equiv 0 \pmod 2 \\
b & \equiv 0 \pmod 3 \\
c & \equiv 42^{-1} \cdot 2022 \equiv 1 \pmod 5 \\
d & \equiv 30^{-1} \cdot 2022 \equiv 3 \pmod 7.
\end{align*}
```

Now set  $a = 2w$ ,  $b = 3x$ ,  $c = 5y+1$ ,  $d = 7z+3$ ; the given equation rewrites as

```
\begin{align*}
2022 & = 105(2w) + 70(3x) + 42(5y+1) + 30(7z+3) \\
& = 210(w+x+y+z) + 132 \\
& \text{iff } 9 = w+x+y+z.
\end{align*}
```

By balls-and-urns the answer is  $\binom{9+3}{3} = 220$ .

## Template #

For a self-contained example, see the [math olympiad proposal submission template](#) that I created for the USEMO.

1. I like the following excerpt from [prettier's documentation on its print-width option](#): “In code style guides, maximum line length rules are often set to 100 or 120. However, when humans write code, they don’t strive to reach the maximum number of columns on every line. Developers often use whitespace to break up long lines for readability. In practice, the average line

length often ends up well below the maximum... Remember, computers are dumb. You need to explicitly tell them what to do, while humans can make their own (implicit) judgements, for example on when to break a line.” ↩

---

[Source repository \(git\)](#) • [Revision history](#) • [Suggest edit](#)

Updated Sat 7 Dec 2024, 21:14:37 UTC by [34be8116b0f9](#)