

Two of the basic processing modes in L^AT_EX are *text mode* and *math mode*. Text mode is processing the paragraph you are now reading; it reads the characters from the input file, generates a glyph (symbol) for each one and builds words, lines and paragraphs from the resulting sequence.

Math mode is used to typeset mathematics; the input is processed differently than text mode. Three key differences:

- Spaces are used for delimiting (ending) commands, but do not generate a space in the rendered output
- Text is italicized, except for certain well-known math terms (*e.g.*, *sin*)
- Certain operators, like `_` and `^` have special meaning

There are several different varieties of math modes; I will show you three of the more common ones here.

The first variety is *inline math mode*. This places a small piece of typeset math within a text paragraph. For example, `$0\le x\le 10$` renders as $0 \leq x \leq 10$ within this paragraph.

Inline math mode begins and ends with the `$` symbol. If you omit either, the L^AT_EX processor will give you an error.

Inline math is good for smaller math items, but note that some math doesn't render very well inline. For example, the summation formula `$\sum_{i=0}^n i = \frac{n(n+1)}{2}$` renders as $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ in inline mode.

The next variety is *display math mode*. The input is the same as inline mode, but the delimiters are `\[` and `\]`. Often, these are on their own line to make reading the input file a bit easier.

Display mode separates the math typesetting apart from the text surrounding it, effectively making the math its own paragraph (which, behind the scenes, is exactly what happens). The main effects, other than creating its own paragraph, are that larger symbols are used for things like summations and integrals, and sufficient vertical space is used to properly render all of the input.

The above summation in display mode renders as:

$$\sum_{i=0}^n i = \frac{n(n+1)}{2}$$

The third variety is a variation of display mode. The *equation* environment allows equations to be numbered for reference elsewhere in the document. The environment begins with `\begin{equation}` and ends with `\end{equation}`. Note that all environments begin and end with `\begin{...}` and `\end{...}`.

The summation formula renders in the *equation* environment as:

$$\sum_{i=0}^n i = \frac{n(n+1)}{2} \tag{1}$$

Note the only difference is the (1) at the right edge. This can be used to refer back to the equation elsewhere in the document.

Although you can manually insert the equation number, the reference feature of L^AT_EX is commonly used to handle numbered equations, algorithms, figures, *etc.* Note that in the source document, I added `\label{eq:sum-of-integers}` after the formula. This creates a *label* that can later be used.

To refer to an equation, use the `\ref{...}` command. To refer to the summation, `\ref{eq:sum-of-integers}` will produce the equation number. However, it only generates the number. Normally, you would use something like `Equation~\ref{eq:sum-of-integers}` to get "Equation 1".

Note the use of a tilde (\sim) instead of a space. This is a *tie* or non-breaking space; it creates a space, but it will force the text to its immediate left and right on the same line. Thus, the word “Equation” and the equation number will always appear on the same line.

Important note: You typically have to run the rendering command twice — and three times for complex documents — to get reference numbering to render properly.

There are thousands of symbols available for both math and text. There are references available online. The Comprehensive T_EX Archive Network keeps an exhaustive list; a 449-page PDF with the list is available at

<https://mirror.math.princeton.edu/pub/CTAN/info/symbols/comprehensive/symbols-letter.pdf>

Here’s a summary of symbols and commands used in this document, other than the math mode delimiters:

- `_` — display an underscore in text mode
- `\^` — display a caret in text mode. Note that it requires `{}` to terminate; otherwise, it will place a circumflex above the next symbol. For example, `\^o` renders as δ .
- `\le` — display a less than or equal to symbol \leq
- `\sum` — display a summation symbol
- `_` — render the next item as a subscript. Two notes:
 - If the item consists of more than one symbol, it must be enclosed in curly braces
 - Depending on context, the subscript may be placed next to the preceding symbol or underneath the symbol. Context involves both the mode — inline or display math mode — and the preceding symbol.
- `^` — render the next item as a superscript. Subscript notes apply here as well.
- `\frac{numerator}{denominator}` — render a fraction with given numerator and denominator
- `\label{...}` — create a label that can be referred to elsewhere in the document.
- `\ref{...}` — refer to a label defined elsewhere in the document. Usually requires two renders to render properly.
- ```` and `''` — render quotation marks at the start and end of quoted text
- `\textasciitilde` — displays a tilde in text mode
- `~` — Creates a tie, which forces the items immediately before and after to appear on the same line with a space between them.
- `\newpage` — ends the current page