

L^AT_EX Do's and Don'ts

1 Modifying Text

When messing with font families:

1. **DO NOT** use the commands `\rm`, `\bf`, `\it`, `\tt` as they are deprecated and low-level T_EX commands. They have syntax which requires you to use curly braces on the outside, thus being easier to use incorrectly, and make errors harder to spot. This is their syntax when used as intended:

```
{\it hello}
```

Instead, **DO** use the commands `\textrm`, `\textbf`, `\textit`, `\texttt` in your code. Example:

```
\textit{hello}
```

Similarly, when changing the font family in mathmode, use `\text` to change something into text in the middle of an environment, and the analogous mathmode commands of the above text commands.

2 Mathematical Writing Quirks

1. **DO NOT** use dollar signs, ever. They are low-level T_EX commands and do not account for certain spacing issues. If you use dollar signs, you will find yourself in certain situations where the spacing around your mathematical writing is incorrect. Most such instances occur when mathmode is used in an argument to the

Instead of using `$MATH$`, **DO** use `\(MATH\)` to wrap your inline mathematics. These are the high-level L^AT_EX version of dollar signs which fix the spacing issues which can occur as a result of using dollar signs.

If you look at this suggestion and are frustrated, thinking that writing two extra characters to switch into mathmode is an absurd proposition, then I suggest using a L^AT_EX editor which supports macros, and learn how to implement and use those macros. I don't ever manually type out `\(\)` to go into mathmode; instead, I have a macro (set to `Ctrl+M`) which types these for me. I suggest everyone who uses L^AT_EX to do the same, since switching between text and mathmode is an extremely common thing to do, and much time will be saved by not having to type the same routine things out by hand over and over.

2. **DO NOT** use the colon symbol `:` on your keyboard to define a function or to specify a type. For example, if you want to define a function f from A to B , do not write

```
\(f: A\rightarrow B\)
```

as the spacing around the colon is, by default, the spacing of a relation symbol. This means that it introduces unwanted spacing in between the name of the function and the colon, like so:

$$f : A \rightarrow B$$

Instead, **DO** use the symbol defined by `\colon` for function definitions. This is the same symbol as the one on your keyboard, but it doesn't introduce the same spacing:

$$f: A \rightarrow B$$

Similarly, to say that a is of type A , one should write `a\colon A` which produces $a: A$, instead of `a:A` which produces $a : A$. To be fair, this second use case is probably more up to preference, whereas when defining functions, the benefits of `\colon` are quite clear.

3. When defining macros for math words, such as the names of maps/functors/categories, like `res`, `Tor`, or **Set**, **DO** use the command `\DeclareMathOperator` as it was specifically defined for this purpose. Examples:

```
\DeclareMathOperator{\res}{res}
\DeclareMathOperator{\Tor}{Tor}
\DeclareMathOperator{\Set}{\mathbf{Set}}
```

4. In case this needs to be said: in general, when defining new macros, **DO NOT** use `\def` ever. This is a low-level deprecated command which should never be used, unless you are writing your own packages, in which case this document is irrelevant to you.

Instead, **DO** use `\newcommand` to define macros and other complex commands.