

# The `mymacros` package

1.0 2021/04/06  
(written 2018 – April 2021)

Andrew Lounsbury

## 1 Description

This package serves as a set of examples of user-defined  $\LaTeX$  commands. Everything here is something that I used or experimented with during my time as an undergraduate student. Some of the commands may not be useful at all.

The primary purpose of most commands in this package is to reduce the amount of typing required.

## 2 Writing Your Own Commands

If you're reading this, you're presumably an undergraduate student who might be interested in using  $\LaTeX$ . In general, I think the best way to start using  $\LaTeX$  and then form a habit of it is to start early and to learn things little by little.

That being said, one of the most useful things you can do is to write commands that make things easier. I would recommend accumulating these commands in one file (package) as you go along, similarly to what I've done here with `mymacros`.

**Example.** In Advanced Calculus I found that I was having to type  $\mathcal{R}(\alpha)$  (`\mathscr{R}(\alpha)`) over and over again. So, I wrote the command

```
\newcommand{\ara}{\mathscr{R}(\alpha)}
```

to avoid this. ‘ar’ is an onomatopoeic rendition of the letter ‘R’ spoken aloud, and the second ‘a’ stands for “alpha.” You have to get creative with naming these things.

## 3 Some Actual Documentation

Some of this information is similar to the formalities you might see in the documentation of other packages.

### 3.1 License

This software is contributed to the public domain. In other words, I don’t care what you do with it. Add things to it, remove things from it, copy things from it to your own set of macros, distribute it, whatever.

### 3.2 Acknowledgements

Most of the `LINEAR ALGEBRA` section—and perhaps a few commands elsewhere—were provided to me in my freshman year by Dr. R. Ablamowicz, emeritus. I’m just going to leave them there.

### 3.3 Dependencies

The `mymacros` package loads `forloop`, `mathrsfs`, `mathtools`, `nicefrac`, `xcolor`, `amsmath`, `amsfonts`, `amssymb`, `amsthm`, `cancel`, `graphicx`, and `verbatim`. Most packages will rely on nowhere near as many packages as this. It only loads this many because `mymacros` is essentially my personal preamble file, so I put whatever I wanted there.

These dependencies may require or include other packages. For instance, the `nicefrac` package includes the `units` package. If you require one of these packages to be loaded with some option, consider loading it yourself before the `mymacros` package or use, e.g.,

```
\PassOptionToPackage[tight]{units}
```

### 3.4 Commands from mathtools

The commands `\Mid` and `\set` are created (and respectively renamed from `\given` and `\Set`) from the documentation of the `mathtools` package as follows:

```
\providecommand\Mid{}
\newcommand\SetSymbol[1][\]{%
  \nonscript\:#1\vert%
  \allowbreak%
  \nonscript\:%
  \mathopen{}}%
}
\DeclarePairedDelimiterX{\set}[1]{\}{\}{%
  \renewcommand\Mid{\SetSymbol[\delimsize]}%
  #1%
}
```

The command `\set` is invoked as `\set{\langle argument \rangle}`:

$$2\mathbb{Z} = \{x \in \mathbb{Z} \mid x \bmod 2 = 0\},$$

but adding a `*` makes everything extensible, in which case we use `\Mid` rather than `\mid`:

$$\mathbb{Q} = \left\{ \frac{x}{y} \mid x, y \in \mathbb{Z} \right\}.$$

### 3.5 For Whom it May Interest

There are a couple of things in `mymacros` that you will likely never use but may be curious about.

#### 3.5.1 Conditionals

At the end of `mymacros.sty` there are some experimental commands for typing ditto marks: `"` `.` They are underdeveloped and not very useful, but I'm leaving them as an example of the more advanced things we can do with `LATEX`, such as for-from-to loops, for-each-in loops, while loops, etc.

One applications of conditionals is in drawing with *TikZ*. For examples of `\foreach`, refer to [this page](#).

I've also used the conditional `\ifdim ... \fi` in my [suppose](#) package.

### 3.5.2 Math Space

The command `\mspace{}`, or math space, is a more precise way of inserting horizontal space in math mode. It accepts rational values of the unit `mu`. The conversion is

$18 \text{ mu} = 1 \text{ \quad} = 1 \text{ em} =$  the width of 'M' in the current font.

$x \quad y$   
 $x \quad y$   
 $xMy$

$1 \text{ \quadquad}$  is the width of 'MM'.

$x \quad y$   
 $x \quad y$   
 $xMMy$

Other spaces are defined to be a certain number of `mu`. For instance, the thin space `\,` is  $3 \text{ mu}$ :

$x y$   
 $x y$

The thick space `\;` is  $6 \text{ mu}$ :

$x y$   
 $x y$

The negative thin space `\!` is  $-3 \text{ mu}$ :

$xy$   
 $xy$