The `alounsburymacros` package

1.0 2022/02/05

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 `alounsburymacros`.

Example. In Advanced Calculus, I found that I was having to type $\mathscr{R}(\alpha)$ (`\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.

I'm not going to take the time to document every command I've made in `alounsburymacros`; however, I've made many comments throughout `alounsburymacros.sty`, If you want to get an idea of what's included in the package, just open the `sty` file in your editor read through those comments.

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.

3.3 Dependencies

The `alounsburymacros` package loads `forloop`, `mathrsfs`, `mathtools`, `nicefrac`, `xcolor`, `amsmath`, `amsfonts`, `amssymb`, `amsthm`, `cancel`, `calligra`, `enumitem`, `fancyvrb`, `graphicx`, `pagecolor`, `pifont`, `suppose`, `totcount`, `upgreek`, `verbatim`, `verse`, and `fontenc`. Most packages will rely on nowhere near as many packages as this. It only loads this many because `alounsburymacros` 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 `alounsburymacros` package or use, e.g.,

```
\PassOptionsToPackage{tight}{units}
```

3.4 Option `b`

Using the `alounsburymacros` package with the option `b`:

```
\usepackage[b]{alounsburymacros}
```

will use the pagecolor package to set the page color to black and the text color to white.

3.5 Commands using mathtools

The commands \Mid and \set are respectively renamed from the commands \given and \Set from page 27 of the mathtools package documentation as follows:

```
% just to make sure it exists
\providecommand\Mid{}
% can be useful to refer to this outside \set
\newcommand\SetSymbol[1][]{%
    \nonscript\:#1\vert%
    \allowbreak%
    \nonscript\:%
    \mathopen{}%
}
\DeclarePairedDelimiterX{\set}[1]\{\}{%
    \renewcommand\Mid{\SetSymbol[\delimsize]}%
    #1%
}
```

The command \set is invoked as \set{⟨*argument*⟩}:

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

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

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

3.6 Proof Environments with the Technique Specified

This package contains proof environments with the technique specified in the title of the proof. I had to create supplementary environments with \newtheorem{...}{...} in order to make the actual environments that have Q.E.D. symbols with \newenvironment{...}{...}. In other words, use

```
\begin{pfco}
    ...
\end{pfco}
```

rather than

```
\begin{pfcombinatorial}
    ...
\end{pfcombinatorial}
```

Proof (Combinatorial). The Q.E.D. symbols in these environments function the same way that they do in the regular `proof` environment. ■

Proof by Induction. We can even push the Q.E.D. symbol into a display that ends the proof:

$$\vec{f}\vec{e_j} = \sum_{i=1}^{\infty} \frac{\partial f_i}{\partial x_j}\vec{u}_j. \qquad \blacksquare$$

3.7 For Whom it May Interest

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

3.7.1 Conditionals

At the end of `alounsburymacros.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 dynamic 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 Ti*k*Z. For examples of `\foreach`, refer to this page.

Assuming I haven't changed it since writing this, I've also used the conditional `\ifnum` ... `\fi` in my `suppose` package.

3.7.2 `\mspace{...}`

The command `\mspace{...}`[1] is a more precise way of inserting horizontal space in math mode. It accepts rational values of the unit `mu`. The conver-

[1] I believe `\mspace{...}` is either short for "`mu` space" or short for "math space," but I'm not sure which.

sion is

18 `mu` = 1 `\quad` = 1 `em` = the width of 'M' in the current font.

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

1 `\qquad` is the width of 'MM'.

$$x \qquad y$$
$$x \qquad y$$
$$xMMy$$

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

$$x\,y$$
$$x\,y$$

The thick space `\;` is 6 `mu`:

$$x\;y$$
$$x\;y$$

The negative thin space `\!` is -3 `mu`:

$$x\!y$$
$$xy$$

Another typesetting unit is `ex`, or x-height, which is the height of the lowercase 'x' character: x.

3.8 `\DeclareMathOperator{...}{...}`

Commands line sin, max, sup are defined as math operators with the command `\DeclareMathOperator{\commandname}{`⟨*text*⟩`}`. It displays ⟨*text*⟩ in

the `mathrm` font and places a space after ⟨*text*⟩ if a delimiter does not immediately follow, which, for instance, is how we write sin by hand when we omit the parentheses.

Compare the following commands that are defined at the beginning of this document:

`\DeclareMathOperator{...}{...}`	`\newcommand{...}{...}`
$\sin \pi$	$\sin\pi$