

How to Speed up Your L^AT_EX Typesetting

Ryan T. Moore*

29 August 2011

1 The Challenge

Typesetting often involves repetitive tasks. You'd probably be surprised how many times you type the same string in a given document (`\begin{itemize}` anyone?). I use Aquamacs' C-c C-e keystroke sequence to invoke environments, but I also use the software-independent solution below.

2 A Solution

First, create new commands in the preamble of your L^AT_EX document using `\newcommand{shortcut}{fullcommand}`, like this:

```
\newcommand{\bi}{\begin{itemize}}
\newcommand{\ei}{\end{itemize}}
```

Be sure that your `shortcut` is not an already-defined L^AT_EX command, or else you will encounter L^AT_EX errors. (If you want to overwrite an already-defined L^AT_EX command, just use `\renewcommand` instead of `\newcommand`.) Then, in your document you need only type

```
\bi
  \item Here's item 1
  \item Here's item 2
\ei
```

This saves many keystrokes for environments like `itemize`, `pmatrix`, and `parbox`. It also works for any string you type frequently. For example, to typeset a bold-faced matrix inverse, \mathbf{A}^{-1} , I define `\ain`, and use four keystrokes instead of 18:

*Robert Wood Johnson Scholar in Health Policy Research, University of California-Berkeley and University of California-San Francisco, 50 University Hall, MC7360, Berkeley CA 94720-7360; Department of Political Science, Washington University in St. Louis, 241 Seigle Hall, Campus Box 1063, One Brookings Drive, St. Louis MO 63130. tel: 510.643.1888; fax: 801.922.7901; rtm (at) berkeley (dot) edu; <http://rtm.wustl.edu>.

```
\newcommand{\ain}{\bf A}^{-1}}
```

Second, you can either include explicit command definitions in the preamble of each document (as just described), or you can use a common set of commands across several documents. To gain the efficiency of the latter, create a file named, e.g., `defs.tex` that stores your frequently-used shortcut commands. Then, insert that document into the preamble of the document on which you're working:

```
\documentclass{article}

\input{defs}

%%% === title, author, etc. ===
\title{Your title}
\author{Your name}
%%% === document ===
\begin{document}
\maketitle
...

\bi
\item Here's item 1
\item Here's item 2
\ei

...

\end{document}
```

3 The Solution, with Arguments

This solution can also take arguments. The full structure of `\newcommand` is

```
\newcommand{name}[number of args][default argument values]{definition}
```

For example, if you declare

```
\newcommand{\mycolors}[2]{Two colors are #1 and #2}
```

then you could use in your document

```
\mycolors{blue}{red}
```

and the compiled result would be “Two colors are blue and red”. For another example, see my “How to Insert BIB_TE_X Entries into a CV, Syllabus, ...”.

Hope this helps!