LaTeX Workshop 3

Dan Parker and David Schwein

5 October 2014

Now that you've been introduced to LaTeX's basic functionality, we're going to explain how to use LaTeX to typeset a certain document with which, for better or for worse, we are all familiar: the problem set. If you use LaTeX for schoolwork, the odds are high that you will use it for making problem sets.

1 Tables

Making a table in LaTeX is just like making a matrix, with a few modifications.

```
\usepackage{booktabs}
...
\begin{center}
\begin{tabular}{ l r }
\toprule
City & Population \\
\midrule
Providence & 178,024 \\
Warwick    &  82,672 \\
Cranston   &  80,387 \\
Pawtucket  &  71,148 \\
\bottomrule
\end{tabular}
\end{center}
```

City	Population
Providence	178,024
Warwick	82,672
Cranston	80,387
Pawtucket	71,148

Let's break down this example. We first load the package `booktabs`, for making typographically good tables. Our table consists of two environments: the `center` environment, which centers the table on the page, and the `tabular` environment, which makes the table proper. The `tabular` environment takes a second argument, in our case `l r`, which tells LaTeX how many columns the table will have and how to align them. The letter `l` is for left-alignment, the letter `r` is for right-alignment, and the letter `c` is for centering. Finally, the commands `\toprule`, `\midrule`, and `\bottomrule` tell LaTeX where to place the horizontal lines separating the descriptors and the columns.

Tables can be very complicated, and for this reason there are many LaTeX packages that provide special table functionality, such as the `colortbl` environment for colored tables. See the LaTeX wikibook for a complete list.

There is a second syntax for making tables, which uses `\hrule` for making horizontal lines and uses vertical bars (the character `|`) in the second argument of the `tabular` environment for making horizontal lines.

```
\usepackage{booktabs}
...
\begin{center}
\begin{tabular}{|l||r|}
\toprule
City & Population \\
\midrule
Providence & 178,024 \\
Warwick    &  82,672 \\
Cranston   &  80,387 \\
Pawtucket  &  71,148 \\
\bottomrule
\end{tabular}
\end{center}
```

City	Population
Providence	178,024
Warwick	82,672
Cranston	80,387
Pawtucket	71,148

It's good to know the second syntax because it is a standard LaTeX feature. We told you about making tables with `booktabs` first because those tables are of much greater quality.[1]

2 Headers

Professors often ask students to write their name on each sheet of a problem set. To do so, and to more generally add text to the header and footer of our LaTeXdocuments, we'll use the `fancyhdr` package. Add the following two lines to your preamble.

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

The second line tells LaTeX to use the new header, the default loaded with the `fancyhdr` package. There are six commands that modify the header.

```
\lhead{Josiah Carberry}
\chead{}
\rhead{Psychoceramics}
\lfoot{}
\cfoot{\thepage}
\rfoot{}
```

The names of the commands are self-explanatory: `\cfoot`, for instance, indicates what goes in the center position of the footer. The command `\thepage` prints the current page number.

[1] Specifically, tables made with `booktabs` are better because they have no vertical lines and few horizontal lines. These characteristics are hallmarks of professional table design.

3 Miscellany

3.1 Adjusting Margin Sizes

The default margins in LaTeX documents are quite wide, and while there is good typographical justification for their size, many people prefer to decrease the margin width. If you are one of these people, we recommend using either of the following packages: the `fullpage` package, which automatically adjusts the margins, or the `geometry` package, which allows finer control over the margins.

`\usepackage{fullpage}` or `\usepackage[margin=2cm]{geometry}`

3.2 Optional Arguments for `\documentclass`

When we told you about the `\documentclass` command, we left out some functionality: `\documentclass` takes optional arguments. For example, writing `\documentclass[twocolumn,12pt]{article}` at the beginning of your TeX file will typeset the document in two columns at 12-point font. See the LaTeX wikibook chapter on document structure for a complete list of optional parameters.

4 Macros

Macros are one of the coolest and most useful features of LaTeX: they are commands that allow you to define other commands! They are very useful time-savers, especially if you find yourself having to type out similar or identical things many times. They are, however, a little bit tricky.

4.1 Macros without Arguments

Let's look at a basic example and then go through it very carefully to see how it works. To start, let's define a command with no arguments.

```
\newcommand{\ftc}{Fundamental Theorem of Calculus}
\ftc

We can then apply the \ftc{} and conclude that...
```

Fundamental Theorem of Calculus
We can then apply the Fundamental Theorem of Calculus and conclude that...

What's going on here? The first line says we want to define a new command whose *name* is `\ftc` and, when that function is used, it should display "Fundamental Theorem of Calculus". Another way to think of this is in terms of copying and pasting. Typing a `\newcommand{\A}{B}` is like saying: copy `B` to memory and, every time you see `\A`, paste in `B`. You can define a `\newcommand` anywhere in the document. However, it will only work *after* the place where you have defined it. It's therefore good style to put any `\newcommand` in the

preamble of the document. Also, one cannot make commands that already have names. If I try to make a newcommand called, say, `\textit`, then an error message will result because this command already exists. There are also some subtle rules about naming commands. As a rule of thumb, user-defined commands should contain only alphabetic characters — numbers and symbols don't work in command names properly.

It's also important to note that, even though our command has no arguments, one should still include empty curly braces after it. Why is this. Try typing the following instead:

```
\newcommand{\ftc}{Fundamental Theorem of Calculus}
\ftc

We can then apply the \ftc and conclude that...
```

Fundamental Theorem of Calculus
We can then apply the Fundamental Theorem of Calculusand conclude that...

As you can see, without the braces, the space is missing after the command. This is because LaTeXassumes that the next character after a command is its first argument, even if the command has no arguments. By explicitly giving the function an empty argument, we can avoid this behavior.

4.2 Macros with Arguments

User-defined commands can have multiple arguments, just like normal commands. `\newcommand` has an optional argument that says how many arguments defined command should have. Suppose we were doing a problem which involved integrating a lot of functions of x over the entire real line. Then it might come in handy to have a command that made an integral from $-\infty$ to ∞ that integrated with respect to x. Again, it's easiest to see how this works by example.

```
\newcommand{\intx}[1]{\int_{-\infty}^\infty #1 \, dx}
\[
\intx{f(x)}
\]
\[
\intx{\frac{3x}{5}e^{-3x}}
\]
```

$$\int_{-\infty}^{\infty} f(x)\,dx$$

$$\int_{-\infty}^{\infty} \frac{3x}{5}e^{-3x}\,dx$$

Even time LaTeXsees a `#1`, it replaces it with the first argument given when the defined command is used. This can appear multiple times inside the command. For instance, we can make a command that repeats something three times:

```
\newcommand{\rep}[1]{#1 #1 #1}
\rep{a} \rep{eggplant}
```

a a a eggplant eggplant eggplant

Making a command that has more than one argument can be done in the same way.

```
\newcommand{\comparison}[2]{I think that #1 is \textit{way} better than #2.}
\comparison{Pepsi}{Coke}
\comparison{Star Trek}{Star Wars}
```

I think that Pepsi is *way* better than Coke.
I think that Star Trek is *way* better than Star Wars.

Hopefully by now you are convinced how cool and how useful macros are. Used carefully and thoughtfully, they can save a great deal of time when writing it LaTeX. If you need to type the same, or a very similar, thing more than three or four times, then it may be worth making it into a macro. Over time, one tends to accumulate a collection of standard macros that you prefer to use. Many people have a few dozen `\newcommand`'s that they put in the preamble of each LaTeX file they make.

5 Useful Packages for Problem Sets

This section will go over a few useful packages for doing things that may turn up in problem sets. Some are broadly useful, and others will only have a use in very specific subjects. There are many more packages for TeXthan can possibly be listed here — over 4000. To learn about all of these, and look through the list, check out `http://www.ctan.org/`.

5.1 The `siunitx` Package

`siunitx` is a package for typesetting units. Units are something of a pain to typeset in normal LaTeX, for a few reasons. Suppose we had measured the frequency of an oscillation to be 3.0×10^{-6}Hz. Using math mode this could be written as `$3.0 \times 10^{-6} \text{Hz}$`. Not only is this long, but the spacing between the number and the unit is terrible! The `siunitx` package offers a better way to do this. Load the package with the command `\usepackage{siunitx}`

`\SI{3e-6}{\hertz}`	$3 \times 10^{-6}\,\mathrm{Hz}$

This is much shorter and faster. The `siunitx` package supports all SI units, as well as quite a few others. It also supports standard prefixes, such as giga- and femto-. Let's look at a few more complex examples.

`\SI{3}{\giga\hertz}`	$3\,\mathrm{GHz}$
`\SI{3}{GHz}`	$3\,\mathrm{GHz}$
`\SI{19}{\meter\per\second}`	$19\,\mathrm{m\,s^{-1}}$
`\SI[per-mode=fraction]{19}{\meter\per\second}`	$19\,\frac{\mathrm{m}}{\mathrm{s}}$
`\SI{54.1}{kg.m/s^2}`	$54.1\,\mathrm{kg\,m/s^2}$
`\SI{32932e-3213}{\micro F}`	$32\,932 \times 10^{-3213}\,\mu\mathrm{F}$
`\SI{3.2}{kg/\nano\meter\squared}`	$3.2\,\mathrm{kg/nm^2}$
`\SI{8.9}{\kilo\gram\tothe{12}}`	$8.9\,\mathrm{kg^{12}}$
`\SI{27}{\degreeCelsius}`	$27\,^{\circ}\mathrm{C}$

There are lots of things going on here. First, you can use either the standard names for units, which all have their own commands, like `\hertz`, or simply type `Hz`. By default, fractions are displayed as negative powers, but you can modify this with the optional argument `[per-mode=fraction]`. If you import the package with the command `\usepackage[per-mode=fraction]{siunitx}`, then all units will be displayed as fractions by default. Lastly, we get access to many base-10 prefixes: kilo-, mega-, giga-, tera-, centi-, milli-, micro-, nano-, pico-, and many more, from 10^{-24} to 10^{24}.

The `siunitx` package contains several other useful commands. `num` works just like the first argument of `\SI` by itself, and `\si` works like the second part.

`\num{3.0}`	3.0
`\num{3.0e5}`	3.0×10^5
`\num{3+2i}`	$3 + 2\mathrm{i}$
`\si{\kilo\gram\per\tesla}`	$\mathrm{kg\,T^{-1}}$
`\si{\lumen\per\radian}`	$\mathrm{lm\,rad^{-1}}$

Additionally, there is a special command for typing numbers with the same units. Numbers are separated by semi-colons, and commands and an "and" are inserted automatically.

`\SIlist{1;2.3;-534;7.3e-5}{\joule\per\second}`
$1\,\mathrm{J\,s^{-1}}$, $2.3\,\mathrm{J\,s^{-1}}$, $-534\,\mathrm{J\,s^{-1}}$ and $7.3 \times 10^{-5}\,\mathrm{J\,s^{-1}}$

For more information on this package, check out it's manual on CTAN: `http://www.ctan.org/pkg/siunitx`.

5.2 The `mhchem` Package

The `mhchem` package serves a similar function with chemical equations. LaTeX's math mode is meant to write mathematics and is thus rather ill-suited for typing chemical equations, mostly because the atomic symbols should not be italicized, and it's a pain to break out of italics all the time. To load this package, use the command `\usepackage{mhchem}`. This allows access to a new command `\ce`, which stands for *chemical equation*. Again, some examples are probably the best way to demonstrate this.

```
\ce{H2O}
\ce{H2SO4}
\ce{NO3-}
\ce{AgCl2-}
\ce{CrO4^2-}
\ce{^{235}_{92}U}
```

H_2O
H_2SO_4
NO_3^-
$AgCl_2^-$
CrO_4^{2-}
${}^{235}_{92}U$

Not only is it possible to do chemical compounds, but also entire chemical equations.

```
\ce{CO2+C -> 2CO}\\[0.4em]
\ce{H+ + OH- <=>> H2O}
```

$CO_2^+C \longrightarrow 2\,CO$
$H^+ + OH^- \rightleftharpoons H_{20}$

This package contains many many more options and variations on this. If you want to learn more, head to the manual at: `http://www.ctan.org/pkg/mhchem`.