Introduction to Scientific Typesetting
Lesson 6: Counters, Theorems, Cross-references, and Lists

Ryan Higginbottom

January 12, 2012

An Overview

Counters

Theorem-like Environments

Cross-referencing

More on Customizing Lists

Counters

What is a Counter?

Counters are what LaTeX uses to keep the right number attached to equations, pages, theorems, etc. Here's a list of the standard counters.

`equation`	`figure`	`footnote`
`page`	`table`	`chapter`
`section`	`subsection`	`subsubsection`
`enumi`	`enumii`	`enumiii`
`enumiv`		

For every counter `name`, there is a command `\thename` which shows the current value of the `name` counter. (We've seen this already when using `\thepage`.)

Setting Counters

The main command here is `\setcounter{`*name*`}{#}`, where # is the value you wish to assign to the counter called `name`.
If you want the first equation in your paper numbered 8 for some reason, you should put this line directly before the `equation` environment:

```
\setcounter{equation}{7}.
```

For most counters, LaTeX increments the counter and then generates the appropriate number. The only exception is with the `page` counter. If you want the first page in your paper to be page 45, you should put this line directly after `\begin{document}`:

```
\setcounter{page}{45}.
```

Advancing Counters

To increment a counter by one, use

```
\stepcounter{name},
```

where `name` is the title of some counter.

There's also the command `\addtocounter{counter}{num}`. Here `num` should be an integer and `counter` should be a recognized counter.

Open the first example file (`.tex`), build to PDF and view.

Let’s practice!

Open the second example file (`.pdf`) from Sakai, and reproduce it.

Theorem-like Environments

What is a Proclamation?

In LaTeX, a *proclamation* is a displayed text environment which contains a theorem, definition, corollary, or something of this sort. These are particular to mathematical writing.

Actually, LaTeX needs to be told what your proclamations are too. For a theorem, you need to put

```
\newtheorem{theorem}{Theorem}
```

in the preamble. Then your theorems go into `theorem` environments.

A Theorem Example

Theorem 1 *In a right triangle with sides a, b, and c, the equation $a^2 + b^2 = c^2$ holds.*

```
\begin{theorem}
In a right triangle with sides $a$, $b$,
and $c$, the equation $a^2+b^2=c^2$ holds.
\end{theorem}
```

There is also an optional way to give names to theorems.

Theorem 2 (The Pythagorean Theorem) *In a right triangle with sides a, b, and c, the equation $a^2 + b^2 = c^2$ holds.*

```
\begin{theorem}[The Pythagorean Theorem]
In a right triangle with sides $a$, $b$,
and $c$, the equation $a^2+b^2=c^2$ holds.
\end{theorem}
```

Defining a Theorem

In the command `\newtheorem{theorem}{Theorem}`, the first argument is the name of the environment and the second argument is the name which is used when LaTeX typesets.

There are three available styles for your proclamations (from most emphatic to least).

`plain`
`definition`
`remark`

Precede the definition of your proclamation with `\theoremstyle{style}` to apply that style to that proclamation.

Note: You will now need the `amsthm` package loaded in your preamble.

First Theorem Example


```
\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem{corollary}{Corollary}

\theoremstyle{definition}
\newtheorem{definition}{Definition}
\newtheorem*{note}{Note}

\theoremstyle{remark}
\newtheorem*{notation}{Notation}
```

Starred versions produce no number.

Open the third example file (`.tex`), build and view.

Numbering Theorems Consecutively

In that last example file, all of the numbered proclamations were numbered with a 1. Some people (including me) don't like this too much, so they use *consecutive numberings*.

```
\newtheorem{theorem}{Theorem}
\newtheorem{lemma}[theorem]{Lemma}
```

The optional argument tells LaTeX to number theorems and lemmas consecutively.

Note: this optional argument must refer to something that has already been defined.

Numbering Theorems Within a Section

The `\newtheorem` command has a second optional argument, which can cause proclamations to be numbered within a section.

For example,
`\newtheorem{proposition}{Proposition}[section]`
causes the propositions in section 1 to be **Proposition 1.1**, **Proposition 1.2**, and so on.

You can combine consecutive numbering and numbering within sections too.

Open the fourth example file (`.tex`), build and view.

The Proof Environment

LaTeX also provides a `proof` environment, used for proofs. It arranges things slightly differently than a theorem or definition, and it ends with the □ symbol.

Add these lines to the end of the last example document.

```
\begin{proof}
This seems fairly obvious.
\end{proof}
```

Build it and view.

A Note for Proclamations and Lists

If a proclamation of yours begins with a list, the spacing doesn't look quite right if you just launch into it. Use `\hfill` directly after opening the environment.

```
\begin{lemma} %\hfill
\begin{enumerate}
\item This is the first fact.
\item This is the second fact.
\end{enumerate}
\end{lemma}
```

Add this to the end of the last example file, build and view. Then, delete the % and do it again to see the difference.

Let's practice!

Open the fifth example file (`.pdf`) from Sakai, and reproduce it.

Cross-referencing

Labels

Within an article, one may wish to refer to another portion of the article, whether a theorem, equation, section, or page. LaTeX makes this easy.

The starting point is to place `\label{name}` at a place to which you'd like to point. Then later in the document typing `\ref{name}` will produce the number you desire.

As an example, if `\label{S:Giraffes}` is placed after the second `\section` command in your document, then `\ref{S:Giraffes}` will produce the number 2.

The arguments of `\label` and `\ref` are case-sensitive.

An Example

Open the sixth example file (`.tex`), build it (twice?) and view.

A few comments:

1. Note that `\eqref` returns the equation number *as it appears in the equation*.
2. Note the use of the tie (tilde) in citing the Theorem.

To see the power of LaTeX, add two lines to your `.tex` file directly before `\begin{theorem}`:

```
\setcounter{theorem}{3}
\setcounter{equation}{20}
```

After numbers change in your article, you don't have to change the references!

Good Labeling Practice

Here are a few hints:

1. Come up with your own, easy-to-remember labeling scheme and be consistent.
2. Place the `\label` command directly after the command that generates the number.

Page References

The command `\pageref{name}` returns the page number where `\label{name}` is located.

Add these two lines to the end of the last example file:

```
\newpage
Go back to page~\pageref{E:area}.
```

Build (twice) and view.

Equation Tags

You can assign a “tag” to an equation by `\tag{text}`, so that *text* will appear on the line with the equation instead of the equation number.

```
\begin{equation}
A = \pi r^2 \tag{area of a circle}
\end{equation}
```

The *text* is typeset as text, so if you wish to use a math symbol, you’ll have to put it in math mode.

```
\begin{equation}
A = \pi r^2 \tag{$\star$}
\end{equation}
```

An Example

Open the seventh example file (`.tex`), build and view.

Notice that the star shows up in the cross-reference too!

References in Multiline Math

You can change the tags in multiline math too. Just place the same `\tag{text}` command on the correct line with the formula whose tag you wish to change.

```
\begin{align}
a &= b \tag{$\bullet$} \\
c &= d
\end{align}
```

Add this to the previous example file, build (twice) and view.

Suppressing Equation Numbers

There are two ways to suppress equation numbers inside of displaymath environments that have numbers.

1. If you want to suppress *all* numbers in that environment, use the starred version of the environment. Use `align*` instead of `align`.
2. If you'd just like to suppress one, use the `\notag` command.

```
\begin{align}
2 \times 3 &= 6 \\
3 \times 4 &= 12 \notag
\end{align}
```

$$2 \times 3 = 6$$
$$3 \times 4 = 12 \tag{1}$$

Let’s practice!

Open the eighth example file (`.pdf`) from Sakai, and reproduce it.

More on Customizing Lists

Customizing Lists

We covered this earlier, but here's a refresher.

```
\begin{itemize}
\item[$\bigstar$] thing one
\item[$\blacktriangledown$] thing two
\end{itemize}
```

- ★ thing one
- ▼ thing two

You'll need the `amssymb` package for these particular symbols. (There's a list of symbols posted on the web site.)

More on `itemize`

Alternatively, you can change these globally, either in the preamble or the body of document:

```
\renewcommand{\labelitemi}{$\star$}.
```

This changes the first level of itemize to use a star instead of a black circle.

Command	Default
`\labelitemi`	•
`\labelitemii`	–
`\labelitemiii`	∗
`\labelitemiv`	·

Example

Try it out! (You'll need the `outlines` package for this.)

```
\renewcommand{\labelitemi}{$\circledast$}
\renewcommand{\labelitemii}{$\blacktriangleright$}
\begin{outline}
\1 thing one
\2 level two
\2 more on this level
\1 back to level one
\end{outline}
```

Add this to the end of your previous example file.

Customizing `enumerate`

Since `enumerate` gives a *numbered list*, there are counters that go along with each of the four levels of possible nesting.

1. First level.
   (a) Second level.
      i. Third level.
         A. Fourth level.

Counters
`enumi`
`enumii`
`enumiii`
`enumiv`

More on `enumerate`

The *label* for each item involves the counter. So the redefinition of labels in `enumerate` is a little more involved than in `itemize`.

1) First level.

   [a] Second level.

```
\renewcommand{\labelenumi}{\arabic{enumi})}
\renewcommand{\labelenumii}{[\alph{enumii}]}
\begin{enumerate}
\item First level.
  \begin{enumerate}
  \item Second level.
  \end{enumerate}
\end{enumerate}
```

Counting in `enumerate`

Here are the possibilities for printing counters:

Command	Result
`\arabic`	arabic numbers
`\alph`	lowercase alphabetic
`\Alph`	uppercase alphabetic
`\roman`	lowercase roman
`\Roman`	uppercase roman

Here are the defaults:

Label	Default	Example
`labelenumi`	`\arabic{enumi}.`	1.
`labelenumii`	`(\alph{enumii})`	(a)
`labelenumiii`	`\roman{enumiii}.`	i.
`labelenumiv`	`\Alph{enumiv}.`	A.

Let's practice!

Open the ninth example file (`.pdf`) from Sakai, and reproduce it.