

---

**Glisterings**

Peter Wilson

Yet in his feverish mind  
 He still could find  
 The miraging domes of Samarkand  
 Glistening through the roiling sand.

---

*Doubt*

The aim of this column is to provide odd hints or small pieces of code that might help in solving a problem or two while hopefully not making things worse through any errors of mine.

Corrections, suggestions, and contributions will always be welcome.

There was an old man named Michael  
 Finnegan  
 Grew some whiskers on his chinnegan.  
 The wind came out and blew them innegan.  
 Poor old Michael Finnegan. Beginnagen.  
 There was ...

---

 TRADITIONAL
**1 Repetition**

There are occasions when an author wants to repeat some text that occurred in a document at some later place, perhaps in an Appendix.

If the original is plain text then this is simple enough — define a macro holding the text and use it in each place the text is to appear. A variation on this is when there is some boilerplate text that will be used in many documents. If the text is short, then define a macro to hold it. If the text is longer, say a page or two, then put the text into a `tex` file and define a macro that `\inputs` that file. I used both these schemes when I was writing class [5] and package [6] files for some ISO documents which tended to have much boilerplate text, both short and long.

Life gets more complicated if you need to repeat something that is automatically numbered by L<sup>A</sup>T<sub>E</sub>X, as requested by David Romano [4] in this partial quote from his posting to `texhax`, which was passed on to me by Barbara Beeton:

*Here is the problem I'm trying to solve. I'd like to have a particular theorem appear twice in a paper, and I'd like its appearance to be identical in both, the numbering in particular, without having to set the section and theorem counters by hand.*

For the following examples I have specified:

```
\newtheorem{again}{Repetitive}[subsection]
```

```
\newsavebox{\Tsaved}
\newcounter{savesub}
\newcounter{savethm}
\newcounter{restoresub}
\newcounter{restorethm}
\newcommand*{\fox}{The slow old fox
  readily leaped over the quick brown dog.}
\newcommand*{\party}{All good men came
  to the party.}
\newcommand*{\boring}{Some things get
  terribly boring when repeated too often.}
```

**1.1 Box it**

Here is one theorem:

**Repetitive 1.1.1** *All good men came to the party.*

and the next one is the one that is to be repeated. One way of repeating some typeset text is to save it in a box and then use that box wherever the text is to be repeated. The typeset theorem (1.1.2) below is produced by the following code:

```
\savebox{\Tsaved}{%
\begin{minipage}{\linewidth}
\begin{again}\label{th1}
\boring
\end{again}
\end{minipage}}
\vspace{\topsep}
\noindent\usebox{\Tsaved}
\vspace{\topsep}
```

**Repetitive 1.1.2** *Some things get terribly boring when repeated too often.*

I discovered that I had to put a vertical space of `\topsep` before and after the box to match the normal spacing around theorems.

And we can use the box again to obtain a repetition of the theorem:

```
\vspace{\topsep}
\noindent\usebox{\Tsaved}
\vspace{\topsep}
```

**Repetitive 1.1.2** *Some things get terribly boring when repeated too often.*

Using this technique a theorem has to be put into a minipage inside the box in order to get the correct line breaking, but minipages don't break across page boundaries. Provided the theorem is short and the stars are aligned then this won't be a problem. On the other hand, for long theorems and normal alignment the repetition will make a page too long.

Here is another theorem (1.1.3) to be repeated. This time, for use later, I save the values of the current `\subsection` and the current `again` theorem counters before typesetting the theorem:

```
\setcounter{savesub}{\value{subsection}}
\setcounter{savethm}{\value{again}}
\begin{again}\label{th2}
\party
\end{again}
```

**Repetitive 1.1.3** *All good men came to the party.*

## 1.2 Saved the numbers

Any `again` theorem in this subsection will normally have a number starting **1.2**.

Having saved the relevant numbers for the first appearance of theorem 1.1.3 we can now typeset it again. The process is:

1. Save the current values of the `\subsection` and the `again` theorem counters.
2. Set the current values of those counters to those for the original `again` theorem.
3. Repeat the theorem.
4. Restore the values of the `\subsection` and the `again` theorem counters.

Here, then, is a repetition of theorem 1.1.3.

```
\setcounter{restoresub}{\value{subsection}}
\setcounter{restorethm}{\value{again}}
\setcounter{subsection}{\value{savesub}}
\setcounter{again}{\value{savethm}}
\begin{again}
\party
\end{again}
\setcounter{section}{\value{restoresub}}
\setcounter{again}{\value{restorethm}}
```

**Repetitive 1.1.3** *All good men came to the party.*

Show another theorem here:

```
\begin{again}
This is a new theorem.
\end{again}
```

**Repetitive 1.2.1** *This is a new theorem.*

And for the third time display theorem 1.1.2:

```
\vspace{\topsep}
\noindent\usebox{\Tsaved}
\vspace{\topsep}
```

**Repetitive 1.1.2** *Some things get terribly boring when repeated too often.*

If the theorem to be repeated is complex then you could either define a macro for it, or put it into a file to be input.

```
\newcommand*{\foxy}{%
\begin{again}
\fox
\end{again}}
\foxy
```

**Repetitive 1.2.2** *The slow old fox readily leaped over the quick brown dog.*

A work that aspires, however humbly,  
to the condition of art should carry its  
justification in every line.

---

*The Nigger of the Narcissus,*  
JOSEPH CONRAD

## 2 Rectangular text

In an earlier column [7] I discussed how to create paragraph shapes of various kinds. One that I had not considered was sought after by Brad Cooper who asked on `comp.text.tex`:

*I am trying to do something ... whereby two lines of large text are justified on the right and left without any hyphenation occurring.*

Several solutions were posted and I give some of them below, in alphabetic order of the respondents.

Donald Arseneau [1] said that the solution was to use a ‘stretch’ `tabular` column type, but that unfortunately there isn’t one. Instead he suggested

```
\noindent\begin{tabular}{@{}r@{}}
\hfilneg A SHORT LINE \\\
\hfilneg A LITTLE LONGER LINE \\\
\hfilneg Donald Arseneau
\end{tabular}
```

which produces:

```
A      SHORT      LINE
A LITTLE LONGER LINE
Donald          Arseneau
```

Or, using the `array` package, like this:

```
\newcolumnntype{s}{>\hfilneg}r}
\enskip\begin{tabular}{s}
A SHORT LINE \\\
...
```

where I have used the `\enskip` space, as well as the regular paragraph indent, to set off the text from the left margin:

```
A      SHORT      LINE
A LITTLE LONGER LINE
Donald          Arseneau
```

Enrico Gregorio [2] posted the following solution, defining a new `center`-like environment:

```
\newenvironment{stretchcenter}%
{$$\let\\\cr\ vbox\bgroup\ialign\bgroup%
```

```
\unskip##\unskip\cr}%
{\crr\egroup\egroup$$$}
```

The `stretchcenter` environment is used just like the regular `center` environment. The result is:

```
A      SHORT      LINE
A LITTLE LONGER LINE
Enrico              Gregorio
```

Dan Luecking [3] posted further solutions. The first simply involves measuring the longest line and putting the others into boxes to match.

```
\newlength\gxx
\settowidth{\gxx}{A LITTLE LONGER LINE}
\noindent\makebox[\gxx][s]{A SHORT LINE}\par
\noindent\mbox{A LITTLE LONGER LINE}\par
\noindent\makebox[\gxx][s]{Dan Luecking}\par
```

This results in:

```
A      SHORT      LINE
A LITTLE LONGER LINE
Dan              Luecking
```

That was the kind of method I had thought of but it does require some manual work. Dan also provided a more elegant solution to match the others, as:

```
\halign{#\cr
A SHORT LINE\cr
A LITTLE LONGER LINE\cr
Dan Luecking\cr}
```

which results in no indentation of the text from the left margin:

```
A      SHORT      LINE
A LITTLE LONGER LINE
Dan              Luecking
```

To have the text indented, add space (`\quad` in the example below) inside the `\halign` like:

```
\halign{\quad#\cr
A SHORT ...}
```

Note that the `\halign` and `\ialign` commands used by Dan and Enrico are usually hidden from  $\text{\LaTeX}$  users but are used by the  $\text{\LaTeX}$  kernel in defining environments like `tabular`, for example.

## References

- [1] Donald Arseneau. Re: Text filling the line. Post to `comp.text.tex` newsgroup, 24 March 2007.
- [2] Enrico Gregorio. Re: Text filling the line. Post to `comp.text.tex` newsgroup, 22 March 2007.
- [3] Dan Luecking. Re: Text filling the line. Post to `comp.text.tex` newsgroup, 23 March 2007.
- [4] David Romano. Setting counters to output of a `\ref` command. Post to `texhax` mailing list, 8 February 2007.
- [5] Peter Wilson.  $\text{\LaTeX}$  for ISO Standards, 2002. [mirror.ctan.org/latex/macros/contrib/isostds/iso](http://mirror.ctan.org/latex/macros/contrib/isostds/iso).
- [6] Peter Wilson.  $\text{\LaTeX}$  Package Files for ISO 10303, 2002. [mirror.ctan.org/latex/macros/contrib/isostds/iso10303](http://mirror.ctan.org/latex/macros/contrib/isostds/iso10303).
- [7] Peter Wilson. Glistings. *TUGboat*, 28(2):229–232, 2007.

◇ Peter Wilson  
18912 8th Ave. SW  
Normandy Park, WA 98166  
USA  
herries dot press (at)  
earthlink dot net