"Hey — it works!"

Jeremy Gibbons
University of Auckland
`jeremy@cs.aukuni.ac.nz`

Welcome to *"Hey — it works!"*. This column is designed for little tricks — even ones that seem trivial or inelegant — that others may find useful. If you have any such items, please send them in to me at the address above. I am particularly looking for short items (up to about a page).

We start off with a little note from our noble editor, describing a trick she used in laying out this issue to get one-column material following two-column on the same page. Then there is a piece from David Lau about 'striking' (drawing a line) through a piece of text, and a piece from Anita Hoover on making rules of different thicknesses in LaTeX alignments. Finally, I describe a modification to LaTeX's `footnote` macro that I used in this issue in order to place verbatim material in footnotes.

One-column material below two-column

Christina Thiele
`cthiele@ccs.carleton.ca`

When you get to the end of this issue, we have an index done with the `\twocolumn` coding described in TTN 1,3:15–16. My problem was to fit a full-width text below the two-column material. I opted for the cheap and sneaky solution: I simply put the text at the end of the file, and then `\hspace`'d it leftwards by -6.5 cm! No argument from TeX, not even a mutter about an overfull hbox.

```
...
\hspace{-6.5cm} {\small
   \begin{tabular}{||p{11cm}||}
      ...
   \end{tabular}
}
\onecolumn
```

Striking through text

David Lau
Richmond, British Columbia
david@worg.questor.org

I present below a macro, `\strikethrough`, which will reproduce its argument with a line through it; for example, '~~strike~~'.

```
\def\strikethrough#1{%
    {%
        \setbox0=\hbox{#1}%
        \dimen0=\ht0 \dimen1=\dp0
        \setbox1=\vbox{
            \box0 \vskip-\dimen1\vskip-0.8ex \hrule
        }%
        \ht1=\dimen0 \dp1=\dimen1
        \box1
    }%
}
```

This macro works by first typesetting the argument, then backing up vertically by an amount equal to its depth plus 0.8 times the x-height of the current font, and finally overlaying a `\hrule` on top of the argument.[1]

Here is the tricky part. It is crucial that the placement of the strikethrough line on the argument does not result in a box (`\box1`) having dimensions different than those of the box (`\box0`) containing the argument by itself. If the superimposition is done imperfectly, the resultant box will not line up properly with the surrounding text. But according to TEX's rules of box construction, `\box1`, as constructed in the code, has a height of $\backslash\mathtt{ht0} - 0.8\,\mathrm{ex} + 0.4\,\mathrm{pt}$ and a depth of $0\,\mathrm{pt}$. As a result, the dimensions of `\box1` have to be adjusted after it has been constructed; which explains why the line "`\ht1=\dimen0 \dp1=\dimen1`" is necessary.

Note that this macro has the shortcoming that it only works for arguments that are relatively "short". In the event that lines of text or entire paragraphs are to be "struck through", a more elaborate approach is called for.

[1] Another approach is to use horizontal leaders, as in Chapter 21 of the TEXbook:

```
\def\strikethrough#1{{%
   \setbox0=\hbox{#1}%
   \dimen0 0.8ex\dimen1\dimen0\advance\dimen1 by0.4pt
   \rlap{\leaders\hrule height \dimen1 depth -\dimen0\hskip\wd0}%
   \box0
}}
```

This involves backing up horizontally, rather than vertically, and using a rule of negative depth. Symbols other than the `\hrule` can be used as leaders. —*jg*

Thicker \hlines in LaTeX alignments

Anita Z. Hoover
University of Delaware
anita@ravel.udel.edu

In the tabular environment, there is a macro called `\hline` which produces a horizontal line in a table across all columns. It is defined to be of a certain thickness and can be changed by modifying the dimension specified by `\arrayrulewidth`. However if you want to keep the original `\hline` thickness of lines, you need to create a new macro, say `\Hline`, to produce thicker lines based on a new dimension, say `\arrayruleHwidth`. You can do this simply by copying the definition for `\hline` and changing the width of the rules. The definition below shows how; it can be placed in the preamble of your document.[2]

```
\newdimen\arrayruleHwidth
\setlength{\arrayruleHwidth}{4pt}
\makeatletter
\def\Hline{\noalign{\ifnum0=`}\fi\hrule \@height \arrayruleHwidth
   \futurelet \@tempa\@xhline}
\makeatother
```

Alternatively, you could omit the `\makeatletter` and `\makeatother` and put the definition in a style file, say `Hline.sty`.

Here is an example of using the new macro `\Hline` to produce thicker lines in the tabular environment.

```
\begin{tabular}{ccc}
   \Hline
   \multicolumn{3}{c}{Very Nice}\\\Hline
   1 & 2 & 3 \\\hline
   4 & 5 & 6 \\\hline
   7 & 8 & 9 \\\Hline
\end{tabular}
```

This produces

Very Nice		
1	2	3
4	5	6
7	8	9

[2] A similar technique could be used for `\vline` and `\cline`. —*jg*

Footnotes with verbatim material

Jeremy Gibbons

There are limitations to what you can do with LaTeX's `\footnote` macro; in particular, you cannot include verbatim material (via `\verb` or `\begin{verbatim}`) in a footnote. This is because of the way LaTeX's footnotes are defined; the `\footnote` macro reads in its entire argument before executing any of it, and the catcodes of the entire argument are fixed when the argument is read—it is not possible to change catcodes within the argument.

On the other hand, plain TeX's footnote macro (as on page 363 of the TeXbook) does not have this problem; it 'uses some `\bgroup` and `\futurelet` and `\aftergroup` trickery, so that the footnote text does not need to be a parameter'.

I needed footnotes with verbatim material for Anita's and David's items in this issue, in order to put editorial comments about their macros in footnotes. I did this by judiciously coalescing LaTeX's and plain TeX's footnote macros, as shown below.

```
% plainfootnote.sty: incorporate plain TeX's trickery into LaTeX's
% footnote macros, to allow \verb"..." within footnotes
% (the argument to \footnote is not read before it is executed)
\long\def\@footnotetext{%
   \insert\footins\bgroup
      \footnotesize
      \interlinepenalty\interfootnotelinepenalty
      \splittopskip\footnotesep
      \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
      \hsize\columnwidth \@parboxrestore
      \edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}%
      \@makefntext{\rule{\z@}{\footnotesep}\ignorespaces}%
      \futurelet\next\fo@t
}
\def\fo@t{\ifcat\bgroup\noexpand\next \let\next\f@@t
                                \else \let\next\f@t\fi \next}
\def\f@@t{\bgroup\aftergroup\@foot\let\next}
\def\f@t#1{#1\@foot}
\def\@foot{\strut\egroup}
```