Glisterings: Glyphs, long labels

Peter Wilson

Ek gret effect men write in place lite;
Th' entente is al, and nat the lettres space.

Troilus and Criseyde, GEOFFREY CHAUCER

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. This installment presents some items about glyphs.

1 Asterism

A symbol called an *asterism* is a simple kind of ornament consisting of three asterisks, and is supplied as a glyph in some fonts. It is typically used as an anonymous division, looking like this:

⁂

Stephen Moye [8] posted a macro to make one if it was not available otherwise. This was based on some earlier code from Peter Flynn [4]. The code below is my version.

```
\newcommand*{\asterism}{%
  \raisebox{-.3em}[1em][0em]{%  OK for 10-12pt
    \setlength{\tabcolsep}{0.05em}%
    \begin{tabular}{@{}cc@{}}%
      \multicolumn{2}{c}*\\[-.75em]%
        *&*%
    \end{tabular}%
  }}
```

The asterism above was printed by:

```
\par{\centering \asterism\par}
```

2 Raising a character

'Maximus_Rumpas' wrote to `ctt` along the following lines:

I am writing some Latin text within a document: GALL : REG : IACO : MAG : BRITA : REG I need to raise the colon between the abbreviated text to the centre of the text line rather than, as normal, aligned at the bottom of the text. I use `\textperiodcentered` *for a single period but I can't find anything similar for colons.*

Heiko Oberdiek [9] responded with:

The following solution centers the colon using an uppercase letter for comparison. Also it makes the colon active inside an environment for easier writing.

```
\documentclass{article}
\begingroup
  \lccode`\~=`\:%
  \lowercase{\endgroup
  \newenvironment{vccolon}{%
    \catcode`\:=\active
    \let~\textcoloncentered
    \ignorespaces
  }{\ifhmode\unskip\fi}}
\newcommand{\textcoloncentered}{}
\DeclareRobustCommand*{\textcoloncentered}{%
  \begingroup
    \sbox0{T}%
    \sbox2{:}%
    \dimen0=\ht0 %
    \advance\dimen0 by -\ht2 %
    \dimen0=.5\dimen0 %
    \raisebox{\dimen0}{:}%
  \endgroup}

\begin{document}
\begin{vccolon}
GALL : REG : IACO : MAG : BRITA : REG
\end{vccolon}
\end{document}
```

Following on from this Dan Luecking suggested using a `\valign`:

```
\def\textcoloncentered{%
  \valign{&##\cr\vphantom{T}\cr\vfil\hbox{:}%
  \vfil\cr}}
```

also remarking that perhaps a simple box raised by some multiple of ex would do as well.

I tried all three suggestions and decided that

```
\DeclareRobustCommand*{\textcoloncentered}{%
  \raisebox{.2ex}{:}}
```

gave a satisfying result, also enabling the height to be adjusted to optically center the colon if necessary..

Heiko's result:

GALL : REG : IACO : MAG : BRITA : REG

Dan's result:

GALL : REG : IACO : MAG : BRITA : REG

My result:

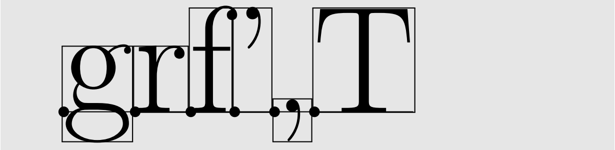
GALL : REG : IACO : MAG : BRITA : REG

3 Boxing a glyph

Paul Kaletta wanted to be able to draw a box around a glyph similar to the example in chapter 11 of *The TEXbook*. Herbert Voß responded with [11] (slightly edited):

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\newsavebox\CBox
\makeatletter
\def\Gbox#1{\begingroup
  \unitlength=1pt
  \fboxsep=0pt\sbox\CBox{#1}%
  \leavevmode
  \put(0,0){\line(1,0){\strip@pt\wd\CBox}}%
  \fbox{#1}%
  \put(-\strip@pt\wd\CBox,0){\circle*{4}}
  \endgroup}
\makeatother
\begin{document}
\begingroup
\fontsize{2cm}{2.2cm}\selectfont
\Gbox{g}\Gbox{r}\Gbox{f}%
\Gbox{'}\Gbox{,}\Gbox{T}
\endgroup
\end{document}
```

Herbert's demonstration code results in:



4 Glyph widths

'PmI' wrote to `ctt`:

I'm trying to put some text in a box, so that I can calculate the box dimensions, but neither hbox nor mbox seem to want to perform linebreaks, and parbox needs a width argument, which is actually what I want to compute so it's useless … example follows:

```
\filltestbox{I need \\ the width \\
             of this text}
\begin{minipage}{\testboxwidth}
```

Several respondents mentioned that the varwidth package [1] does this, but two other solutions were provided for when the text was simple [3].

Ulrike Fischer proposed using a tabular and measuring its width.

```
\newsavebox\testbox
\newcommand{\filltestbox}[1]{%
  \savebox\testbox{%
    \begin{tabular}{@{}l@{}}#1\end{tabular}}}
\newcommand*{\testboxwidth}{\the\wd\testbox}
```

Alternatively, as an exercise Donald Arseneau, the author of the varwidth package, proposed this:

```
\newcommand{\filltestbox}[1]{%
  \setbox\testbox\vbox{%
  \def\\{\unskip\egroup\hbox\bgroup%
         \ignorespaces}%
  \hbox\bgroup\ignorespaces #1\unskip\egroup}}
```

which makes a vbox (called `\testbox`) containing a list of hboxes whose contents are the pieces of text between \\.

In my limited testing, the two approaches yield the same final result. For instance:

```
\filltestbox{Some \\
             text of which I want to know \\
             the width. \\
             There can be \\
             only one paragraph.}
\fbox{%
\begin{minipage}{\testboxwidth}
\usebox\testbox
\end{minipage}}
```

results in:

Some
text of which I want to know
the width.
There can be
only one paragraph.

5 Font size

Gonzalo Medina Arellano asked on `ctt`:

Let's say I use 12pt as a class option. How can I find the exact values of the size obtained with the standard commands `\tiny`*,* `\scriptsize`*, …,* `\huge`*, and* `\Huge`*?*

Several respondents, Bob Tennent [10] among them, suggested looking at the appropriate `.clo` file, such as `size12.clo` for the article or report classes' `12pt` option or `bk12.clo` for the book class, which lists the font and baseline sizes for the several size commands.

Dan Luecking [6] added to this, saying (edited a bit):

You can determine the current font size within a document without knowing what size command was last issued. The macro `\f@size` *holds the font size [as a number (10, 12, 14.4, …)] and is updated with each size change. For convenience we can define another macro without an* `@` *sign to access it, as in:*

```
\makeatletter
  \newcommand*{\currentfontsize}{\f@size}
\makeatother
```

Then `\currentfontsize` *would print it and*
`\typeout{\currentfontsize}`
would display it on the terminal screen and in the log file.

It turns out that the font sizes are the same in `size*.clo` and `book*clo` (the differences are in various margin settings). The unofficial LaTeX reference manual provides a table of the font sizes at [5].

> I don't believe in labels. I want to do the best I can, all the time. I want to be progressive without getting both feet off the ground at the same time.
>
> *Television and radio interview, March 15, 1964*, Lyndon B. Johnson

6 Long labels

Ernest posted to `comp.text.tex` saying [2]:

> *I'm trying to change the description environment, so that when labels exceed a certain length, the text following the label starts in the next line instead of starting in the same line. The LaTeX Companion book explains how to do this, however, when I've tried I've found that the lines that contain a long label are typeset a little bit too close to the previous line, not with the usual baselineskip ...*

The standard `description` environment uses `\descriptionlabel` for setting the contents of the `\item` macro. The *Companion* [7, §3.3] describes various methods of modifying the standard layout by using a different definition for `\descriptionlabel` and/or creating a new kind of `description` list. Ernest's requirement can be met by a modified version of `\descriptionlabel`, the default definition of which is:

```
\newcommand*{\descriptionlabel}[1]{%
  \hspace{\labelsep}%
  \normalfont\bfseries #1}
```

The following is an example of the default appearance of a `description` list:

Short A short label.

Longer label A longer label.

A very long label exceeding the available width
A long label with the text also longer than a single line.

Medium label The more typical length of a label and some text.

As you can easily see, it does not handle long `\item` labels in a graceful manner.

This version of the `\descriptionlabel` meets Ernest's requirements:

```
\usepackage{calc} % or xparse
\newlength{\dlabwidth}
\newcommand*{\widedesclabel}[1]{%
  \settowidth{\dlabwidth}{\textbf{#1}}%
  \hspace{\labelsep}%
  \ifdim\dlabwidth>\columnwidth
    \parbox{\columnwidth-\labelsep}%
      {\textbf{#1}\strut}%
  \else
    \textbf{#1}
  \fi}
\let\descriptionlabel\widedesclabel
```

which, when applied to the previous example yields:

Short A short label.

Longer label A longer label.

A very long label exceeding the available width
A long label with the text also longer than a single line.

Medium label The more typical length of a label and some text.

References

[1] Donald Arseneau. The `varwidth` package, March 2009. `http://mirror.ctan.org/macros/latex/contrib/varwidth`.

[2] Ernest. Description environment. Post to `comp.text.tex` newsgroup, 17 June 2010.

[3] Ulrike Fischer and Donald Arseneau. Re: line breaks in boxes (or 'how do i get paragraph parsing in hbox/mbox?'). Post to `comp.text.tex` newsgroup, 26–27 November 2009.

[4] Peter Flynn. Re: Uncommon typography. Post to `comp.text.tex` newsgroup, 4 June 2007.

[5] George Greenwade et al. LaTeX: An unofficial reference manual. `http://svn.gna.org/viewcvs/*checkout*/latexrefman/trunk/latex2e.html#Font-styles`. Package home page: `http://home.gna.org/latexrefman`.

[6] Dan Luecking. Re: How to find the fontsize? Post to `comp.text.tex` newsgroup, 24 August 2010.

[7] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. Addison Wesley, second edition, 2004. ISBN 0-201-36299-6.

[8] Stephen Moye. Re: asterism. Post to `xetex` mailing list, 11 January 2010.

[9] Heiko Oberdiek. Re: Raising a colon to center. Post to `comp.text.tex` newsgroup, 23 November 2009.

[10] Bob Tennent. Re: How to find the fontsize? Post to `comp.text.tex` newsgroup, 24 August 2010.

[11] Herbert Voß. Re: How to draw a box around the boundaries of a glyph? Post to `comp.text.tex` newsgroup, 27 March 2009.

◇ Peter Wilson
12 Sovereign Close
Kenilworth, CV8 1SQ
UK
`herries dot press (at) earthlink dot net`