The fonttable package

Author: Peter Wilson, Herries Press
Maintainer: Will Robertson
will dot robertson at latex-project dot org

v1.6c 2017/06/06

Abstract
The package lets you typeset the characters in a font in tabular and/or running text forms.

Contents
1 Introduction 1
2 The package 1
  2.1 Table and texts ................. 2
  2.2 Testing a glyph ................. 4
3 The code 5
  3.1 Table and texts ................ 5
  3.2 Testing a glyph ................. 13

Bibliography 15

1 Introduction
The fonttable package lets you typeset a font’s character set in tabular and/or running text forms.

This manual is typeset according to the conventions of the \TeX{} doctest utility which enables the automatic extraction of the \TeX{} macro source files [MG04].

2 The package
The package provides commands to typeset a table of all the glyphs in a given font and to typeset an example of regular text. For font designers it provides commands to typeset a ‘test’ glyph among sets of glyphs from the font.
As a convenience, \fnthours prints the time of day when the file was processed; it uses the 24 hour clock notation. (The macro \today prints the date when the file was processed.)

\section{Table and texts}

The command \fonttable{⟨testfont⟩}

\begin{verbatim}
\fonttable{(testfont)}
\end{verbatim}

typesets a table showing all the glyphs in the ⟨testfont⟩, where ⟨testfont⟩ is the name of a font file\footnote{More precisely, the name of a .tfm file.} like \texttt{cmr10} (for Computer Modern Roman) or \texttt{pzdr} (for Zapf Dingbats).

NOTE: The mfinc package \cite{Pak05} for pretty-printing METAFONT code also defines a \fonttable macro that is akin to this one. If you want to use both packages together then you can use the following general procedure for when a macro \macro is defined in both packA and packB packages.

\begin{verbatim}
\usepackage{packA}
\let\macroA\macro% save packA’s definition
\let\macro\relax% undefine \macro
\usepackage{packB}% now it’s packB’s definition of \macro
...
\macro % use the packB definition
\macroA % use the packA definition
\end{verbatim}

\xfonttable{⟨encoding⟩}{⟨family⟩}{⟨series⟩}{⟨shape⟩}

\begin{verbatim}
\xfonttable{(encoding)}{(family)}{(series)}{(shape)}
\end{verbatim}

typesets a table showing all the glyphs in the font with encoding ⟨encoding⟩ (e.g., T1 or OMS), family ⟨family⟩ (e.g., ppl for Palatino or \texttt{cmbrs} for CM Bright Math (OMS)), font series ⟨series⟩ (e.g., sb for semibold of m for medium), and font shape ⟨shape⟩ (e.g., n for normal or sc for small caps). For example:

\begin{verbatim}
\xfonttable{T1}{ppl}{m}{n}
\end{verbatim}

for Zapf Dingbats.

The command\footnote{The name was chosen in an attempt to avoid clashes with other macros that might perform similar functions.}

\pikfont{⟨encoding⟩}{⟨family⟩}{⟨series⟩}{⟨shape⟩}

\begin{verbatim}
\pikfont{(encoding)}{(family)}{(series)}{(shape)}
\end{verbatim}

selects the font with encoding ⟨encoding⟩ (e.g., T1 or OMS), family ⟨family⟩ (e.g., ppl for Palatino or \texttt{cmbrs} for CM Bright Math (OMS)), font series ⟨series⟩ (e.g., sb for semibold of m for medium), and font shape ⟨shape⟩ (e.g., n for normal or sc for small caps). For example:

\begin{verbatim}
\pikfont{T1}{ppl}{m}{sc}
\end{verbatim}

for Palatino small caps. The size of the font corresponds to the current setting (e.g., \footnotesize, \normalsize, \Large). It can also be changed after being selected by the incantation

\begin{verbatim}
\fontsize{(size)}{(baselineskip)}\selectfont
\end{verbatim}
2.1 Table and texts

where \( \langle \text{size} \rangle \) is the normal height and \( \langle \text{baselineskip} \rangle \) is the distance between text lines; the measurement system is \text{pts} but just use numbers with no units specified. For example:

\[
\text{\fontsize{12}{15}\selectfont}
\]

for a 12pt font with 15pts between baselines.

If you are unsure about the meaning of the various arguments of \texttt{xfonttable} and \texttt{pikfont} see The Companion [MG04, Chapter 7] or the LaTeX2e font selection manual (\texttt{fntguide.tex}; try \texttt{texdoc fntguide}).

\texttt{\fontrange{\langle{\text{low}}\rangle}{\langle{\text{high}}\rangle}}

The package attempts to populate the table with a maximum of 256 glyphs, numbered from 0 to 255. The \texttt{\fontrange{\langle{\text{low}}\rangle}{\langle{\text{high}}\rangle}} declaration changes this by reducing the range so that it extends from \( \langle{\text{low}}\rangle \) to \( \langle{\text{high}}\rangle \), where \( \langle{\text{low}}\rangle \) should be at least 0 and \( \langle{\text{high}}\rangle \) at most 256, and \( \langle{\text{low}}\rangle \) less than \( \langle{\text{high}}\rangle \).

The table is composed of blocks of sixteen characters. If necessary the value of \( \langle{\text{low}}\rangle \) is adjusted lower and \( \langle{\text{high}}\rangle \) is adjusted higher to match this block structure. For example, if you wanted a table of the lower 128 characters then \texttt{\fontrange{0}{127}} would do the job, while the upper half of a 256 character font could be tabulated via \texttt{\fontrange{128}{255}}.

\texttt{\decimals} \hspace{1em} \texttt{\nodecimals}

Normally each cell in the table includes the decimal number of the position in the (256) character set. \texttt{\nodecimals} turns off this numbering and \texttt{\decimals} turns it on. The default is \texttt{\decimals}.

\texttt{\hexoct} \hspace{1em} \texttt{\nohexoct}

Normally the columns and rows in the table are numbered using hexadecimal and octal numbers. These can be turned off by \texttt{\nohexoct} and turned on again with \texttt{\hexoct}, which is the default.

\texttt{\ftablewidth}

The font table’s width is the length \texttt{\ftablewidth}, which by default is set to the normal textwidth (or more exactly, to \texttt{\hsize}). The table itself is left aligned. However, if \texttt{\nohexoct} is in effect the width of the table is its natural width.

\texttt{\fntcolwidth}

When \texttt{\nohexoct} is in effect the minimum width of a table column is \texttt{\fntcolwidth}. This is initially declared as \texttt{\setwidth{\fntcolwidth}{0.08\ftablewidth}}

\texttt{\fonttext}

The command \texttt{\fonttext{\langle{\text{testfont}}\rangle}} typesets an example text using the \langle{\text{testfont}}\rangle (e.g., \texttt{\cmr\rmfamily}).

\texttt{\simpletext} \hspace{1em} \texttt{\fulltext}

The example text can be just a paragraph and a line of capitals, or include more complex accented words as well. Following the declaration \texttt{\fulltext} the complex words are included as well as the example paragraph. The default is \texttt{\simpletext} for just the paragraph.

\texttt{\regulartext}

The command \texttt{\regulartext{\langle{\text{fontspec}}\rangle}} typesets the example text using \langle{\text{fontspec}}\rangle, for example \texttt{\rmfamily\itshape} or \texttt{\pikfont{T1}\{pnc}\{m\}\{it\}}.

\texttt{\fonttexts} \hspace{1em} \texttt{\regulartexts}

The macro \texttt{\fonttexts{\langle{\text{testfont}}\rangle}{\langle{\text{text}}\rangle}} typesets \langle{\text{text}}\rangle using the \langle{\text{testfont}}\rangle \texttt{\regulartexts} (e.g., \texttt{\rmfamily\itshape}). Similarly the macro \texttt{\regulartexts{\langle{\text{fontspec}}\rangle}{\langle{\text{text}}\rangle}} typesets \langle{\text{text}}\rangle using \langle{\text{fontspec}}\rangle (e.g., \texttt{\rmfamily\itshape} or \texttt{\pikfont{T1}\{ppl\}\{m\}\{it\}}).

\texttt{\germanparatext} \hspace{1em} \texttt{\latinparatext}

\texttt{\germanparatext} expands to a German language paragraph, borrowed from the \texttt{blindtext} package [Lik05]. \texttt{\latinparatext} expands to one version of a paragraph of the traditional \texttt{lorem ipsum} dummy Latin text. Either, or both, of these could be used as the \langle{\text{text}}\rangle argument to \texttt{\fonttexts} or \texttt{\regulartexts}.

NOTE: These were originally called \texttt{\germantext} and \texttt{\latintext} but on 2009/05/14 I was told that the babel package defines \texttt{\latintext}, which causes
The package produces unexpected results if it is used in the same document as this package. To try and be on the safe side I renamed \germantext as well as \latintext.

\aztext \aztext expands to the lowercase Latin alphabet a to z, and \AZtext is the corresponding command for the uppercase A to Z. The macros \digitstext and \punctext expand respectively to the digits 0 to 9, and to the typical punctuation marks. In all cases there is a space between each character.

\aztext \aztext expands to the lowercase Latin alphabet a to z, and \AZtext is the corresponding command for the uppercase A to Z. The macros \digitstext and \punctext expand respectively to the digits 0 to 9, and to the typical punctuation marks. In all cases there is a space between each character.

2.2 Testing a glyph

The macros here are a reimplementation of Donald Knuth’s testfont.tex, which is available from CTAN.

In the following, the value of a glyph argument can be specified as its location in the font (i.e., as a decimal number). With a few exceptions, if the glyph is within the visible ASCII range (33–126) it may instead be specified by the ASCII character prefixed with a single open quote mark \textsuperscript{3} ('). The exceptions are nos: 37 (‘), 92 (\) 123 (}) and 125 (}) (but there may be others). In any case, the glyph representing the character p can be specified either as ’p or as 112.

The glyphs are taken from the current font. If the font does not have Latin alphabet glyphs in the ASCII locations then in the descriptions below phrases like ‘lowercase alphabet’ or ‘uppercase alphabet’ or ‘digits’, should be taken to mean (the glyphs in) those locations.

\glyphmixture \ glyphmixture\{(T)\}{(S)}\{(E)\} typesets the \(T\) (test) glyph between the glyphs in the range from \(S\) (start) to \(E\) (end). For example
\glyphmixture\{’e\}{’f\}{’g\} will produce
efeefeeeffef

glyphmixture\{’e\}{’f\}{’g\} will produce
efeefeeeffef

glyphalternation \ glyphalternation\{(T)\}{(S)}\{(E)\} typesets the \(T\) glyph alternately between each glyph in the range from \(S\) to \(E\). For example
\glyphalternation\{’e\}{’f\}{’g\} will produce
efefefefefefefef

glyphseries \ glyphseries\{(T)\}{(S)}\{(E)\} typesets the \(T\) glyph between the glyphs in the range from \(S\) to \(E\). For example
\glyphseries\{’e\}{’f\}{’h\} will produce
efehe

\glyphalphabet \ glyphalphabet\{(T)\} typesets the \(T\) glyph between each letter of the lowercase Latin alphabet plus a few others. \GLYPHALPHABET\{(T)\} does the same but using the uppercase Latin alphabet. For example, the output of \glyphalphabet\{’3\} is like
3a3b3c3d3e3f3g...3a303^313”3

\glyphlowers \ glyphlowers takes each character of the lowercase alphabet in turn as a test glyph and sets it interspersed among the other lowercase characters.
\glyphdigitstext and \punctext are similar except that they use the uppercase alphabet and the ten digits instead. For example, \glyphdigits produces

\textsuperscript{3}Sometimes called a ‘backquote’.
\texttt{\textbackslash glyphpunct} sets a collection of words with an assortment of punctuation marks.

3 The code

1 (*pack*)

3.1 Table and texts

Most of the code below is an edited version of code used in \texttt{nfssfont.tex} for displaying aspects of the set of glyphs in a font.

\texttt{\sevenrm} A small fixed size roman font.

\texttt{\fonttable} typesets a table of all the glyphs in the \texttt{\langle font \rangle} (e.g., auncl10).

\texttt{\pikfont} selects the font with \texttt{\langle encoding \rangle}, \texttt{\langle family \rangle}, \texttt{\langle series \rangle} and \texttt{\langle shape \rangle}.

\texttt{\xfonttable} typesets a table of all the glyphs in the font with \texttt{\langle encoding \rangle}, \texttt{\langle family \rangle}, \texttt{\langle series \rangle} and \texttt{\langle shape \rangle} (e.g., \texttt{\xfonttable{T1}{pnc}{m}{it}} for New Century Schoolbook italic). The original code for the macro was supplied by Enrico Gregorio.

output like

000102030405060708090
101112131415161718191
202122232425262728292
... 
90192939495969798999
New: strip any size information from the fontname (which could be, e.g., either ‘cmr10’ or ‘cmr10\ at\ 10pt’.) This wasn’t necessary before because we didn’t explicitly choose the font size; it was inferred automatically.

\edef\@tempa{\string a\string t}\%
\edef\@tempb{\noexpand\in@{\@tempa}{\f@tfontname}}\%
\@tempb
\ifin@
\edef\f@tfontname{\expandafter\f@tstripsize\f@tfontname}\%
\fi
End new code, and finish as before:
\normalfont
/f@tstartfont
/ftable
/endgroup
}

\f@tstripsize Needed above.
\edef\@tempa{\def\noexpand\f@tstripsize##1 \string a\string t##2 \string p\string t{##1}}\%
\@tempa
\f@tstartfont
Sets up for a font table.
\newcommand*{\f@tstartfont}{%
New: scale the font by 0.01% to (attempt to) avoid TeX’s font optimisation. This becomes a problem in Spanish babel, say, when \textfont/f@textfont changes when cmr10 has been loaded under a different name, here. (And the \textfont can no longer be parsed correctly. See: http://latex-alive.tumblr.com/post/3229118083/texs-font-loading-optimisation)
\@tempdima=\f@size pt
\font\f@ttestfont=\f@tfontname at 0.9999\@tempdima\relax
Continue as before:
/\f@ttestfont \f@tsetbaselineskip
/\ifdim\fontdimen6\f@ttestfont<10pt\relax
/\rightskip=0pt plus 20pt\relax
/\else
/\rightskip=0pt plus 2em\relax
/\fi
/\spaceskip=\fontdimen2\f@ttestfont % space between words (\raggedright)
/\xspaceskip=\fontdimen2\f@ttestfont
/\advance\xspaceskip by\fontdimen7\f@ttestfont
/\f@tsetbaselineskip
\newcommand*{\f@tsetbaselineskip}{\setbox0=\hbox{\f@tn=0
/\loop\char\f@tn \ifnum \f@tn<255 \advance\f@tn 1 \repeat}
3.1 Table and texts

50 \baselineskip=6pt \advance\baselineskip\ht0 \advance\baselineskip\dp0 }

51 \f@oct\f@oct\{\onum\} typesets the octal constant \langle\onum\rangle.
52 \newcommand*{\f@oct}[1]\{\hbox{\rmfamily'\kern-.2em\itshape #1\kern.05em}} % octal constant

53 \f@thex\f@thex\{\hnum\} typesets the hexadecimal constant \langle\hnum\rangle.
54 \newcommand*{\f@thex}[1]\{\hbox{\rmfamily\H{}\ttfamily#1}} % hexadecimal constant

55 \def\f@tsetdigs#1"#2{\gdef\h{#2}% \h=hex prefix; \0\1=corresponding octal
56 \f@tm=f@tm \divide\f@tm by 64 \xdef\0{\the\f@tm}%
57 \multiply\f@tm by-64 \advance\f@tm by\f@tn \divide\f@tm by 8 \xdef\1{\the\f@tm}}

58 \f@ttestrow\f@ttestrow checks if there are any characters in the next block of 16 slots.
59 \newcommand*{\f@ttestrow}\{\setbox0=\hbox{\penalty 1\def\char"h}\h} % \f@tp=1 if none of the characters exist
60 \global\f@tp=lastpenalty} % \f@tp=1 if none of the characters exist
61 \ifhexoct
62 \newif\ifhexoct
63 \newcommand*{\hexoct}{\hexocttrue}
64 \newcommand*{\nohexoct}{\hexoctfalse}
65 \hexoct
66

67 \f@toddlinenum \f@toddline
68 \newcommand*{\f@toddline}\{\cr
69 \noalign{\nointerlineskip}
70 \multispan{19}\hrulefill&
71 \setbox0=\hbox{\lower 2.3pt\hbox{\f@thex\{\h x\}}}\smash{\box0}
72 \cr
73 \noalign{\nointerlineskip}}

74 \iff@tskipping
75 \newif\iff@tskipping
76 \f@tskippingtrue
77 \f@tskippingfalse
78 \fontrange\fontrange\{\langle\low\rangle\}\{\langle\high\rangle\} sets the character range to be output.
79 \newcommand*{\fontrange}[2]\{% 80 \ifnum#1<#2\relax
81 Set \f@tlow to the nearest multiple of 16 that is at or below \langle\low\rangle, but first make
82 sure that it will be at least 0.
83 \ifnum#1<\z@ \f@tm=\z@
84 \f@tlow=\z@\}
85 %
86 %
\begin{verbatim}
\else
  \f@tm=#1
  \divide \f@tm \sixt@@n
  \multiply \f@tm \sixt@@n
  \fi
  \edef\f@tlow{\the\f@tm}

Set \f@thigh to the nearest multiple of 16 at or above \textit{(high)}, finally making sure
that its maximum is 256.

\f@tm=#2
  \divide \f@tm \sixt@@n
  \advance \f@tm \@ne
  \multiply \f@tm \sixt@@n
  \ifnum \f@tm > \@cclvi \f@tm=\@cclvi \fi
  \edef\f@thigh{\the\f@tm}
  \else
  \PackageError{fonttable}{Improper values for fontrange. Default values substituted}{\@ehc}
  \def\f@tlow{0} \def\f@thigh{256}
  \fi
  \fontrange{0}{256}

\f@tloopforsixteen \f@tloopforsixteen sets up a block of sixteen character slots.
\begin{verbatim}
\newcommand*{\f@tloopforsixteen}{
  \ifnum\f@tn<\f@tlow \global\f@tn=\f@tlow\fi
  \loop\f@tskippingfalse
  \ifnum\f@tn<\f@thigh \f@tm=\f@tn \divide\f@tm \sixt@@n \chardef\next=\f@tm
  \expandafter\f@tsetdigs\meaning\next \f@ttestrow
  \ifnum\f@tp=\@ne \f@tskippingtrue \fi\fi
  \iff@tskipping \global\advance\f@tn \sixt@@n \repeat

\f@tevenline \f@tevenlinenonum \f@tevenline gets next non-empty set of a block of 16 characters. It either calls
\f@tmorechart to print them, or \f@tendchart to finish off the table if all 256
potential characters have been processed.
\f@tevenlinenonum does something similar when no external numbers are
printed.
\newcommand*{\f@tevenline}{
  \f@tloopforsixteen
  \ifnum\f@tn=\f@thigh \let\next=\f@tendchart\else\let\next=\f@tmorechart\fi
  \next}
\newcommand*{\f@tevenlinenonum}{
  \f@tloopforsixteen
  \ifnum\f@tn=\f@thigh
    \\\hline
  \else
    \\\hline
  \f@tmorechartnonum
  \f@tmorechartnonum
  \fi}
\end{verbatim}
\end{verbatim}
\texttt{\textbackslash ftmorechart} \texttt{\textbackslash ftmorechart} sets two lines of the table, and \texttt{\textbackslash ftmorechartnonum} does the same when there are no external numbers.

\texttt{\textbackslash ftchartline} \texttt{\textbackslash ftchartline} does a line of the table, including external numbers, and \texttt{\textbackslash ftsimpleline} \texttt{\textbackslash ftsimpleline} does an unnumbered line.

\texttt{\textbackslash ftchartstrut} \texttt{\textbackslash ftchartstrut} is a strut used in each table line. \texttt{\textbackslash fttablewidth} \texttt{\textbackslash fttablewidth} is width of an externally numbered table. \texttt{\textbackslash fntcolwidth} \texttt{\textbackslash fntcolwidth} is the minimum width of a column in an unnumbered table.

\texttt{\textbackslash ftcol} \texttt{\textbackslash ftcol} is a table line of spaces, with no verticals.

\texttt{\textbackslash ftstartchartnonum} \texttt{\textbackslash ftstartchartnonum} is a table line of spaces, with no verticals.

\texttt{\textbackslash ftable} \texttt{\textbackslash ftable} sets a complete character table. The actual code is in either \texttt{\textbackslash fttablenum} or \texttt{\textbackslash fttablenum} for externally numbered or plain tables, respectively.
\f@tendchart \f@tendchart sets the last line of an externally numbered table with the relevant hex digits.

\newcommand*{\f@tendchart}{\cr
noalign{\hrule}
\raise11.5pt\null&&&\f@thex 8&&\f@thex 9&&\f@thex A&&\f@thex B&
&\f@thex C&&\f@thex D&&\f@thex E&&\f@thex F&\cr
\egroup$$\par}
\f@tpsg \f@tpsg typesets a single glyph, possibly with its decimal slot number. \f@placechar is the function to typeset the glyph with its number that is internally defined as \f@placedecimal if decimals are to be shown.

\newcommand*{\f@placechar}{\@firstoftwo}
\newcommand*{\f@placedecimal}{\f@placedecimal}
\newcommand*{\f@treposition}{\setbox0=\vbox{\kern2pt\box0}\f@tdim=\dp0
\advance\f@tdim 2pt \dp0=\f@tdim}

\decimals Following \decimals, which is the default, decimal numbers are printed in the table. Following \nodecimals they are not printed.

\newcommand*{\nodecimals}{\renewcommand*{\f@placechar}{\@firstoftwo}}
\newcommand*{\decimals}{\renewcommand*{\f@placechar}{\f@placedecimal}}

\f@treposition \f@treposition
\newcommand*{\f@treposition}{\setbox0=\vbox{\kern2pt\box0}\f@tdim=\dp0
\advance\f@tdim 2pt \dp0=\f@tdim}
3.1  Table and texts

\fonttext \fonttext{(font)} typesets \knutext using (font) (e.g. auncl10).
\def\fonttext#1{%  
\def\f@tfontname{#1}%  
\bgroup \f@tstartfont \knutext \egroup}

\regulartext \regulartext{(fontspec)} typesets \knutext using (fontspec) (e.g., \aunclfam).
\def\regulartext#1{%  
\bgroup \#1 \knutext \egroup}

\knutext Deathless prose from Knuth for testing a font. It includes \moreknutext, \capknutext, and \knunames.
\def\knutext{{  
On November 14, 1885, Senator \& Mrs. Leland Stanford called together at their San Francisco mansion the 24 prominent men who had been chosen as the first trustees of The Leland Stanford Junior University.
They handed to the board the Founding Grant of the University, which they had executed three days before. This document---with various amendments, legislative acts, and court decrees---remains as the University’s charter. In bold, sweeping language it stipulates that the objectives of the University are ‘‘to qualify students for personal success and direct usefulness in life; and to promote the publick welfare by exercising an influence in behalf of humanity and civilization, teaching the blessings of liberty regulated by law, and inculcating love and reverence for the great principles of government as derived from the inalienable rights of man to life, liberty, and the pursuit of happiness.’’
\moreknutext
\capknutext
\knunames
\par}}

Some more text with a variety of ligatures and accents.
\def\moreknutext{'But aren’t Kafka’s Schlo\{ss\} and \{\AE\}sop’s \OEuvres often na\"i\}ve vis-\'a-vis the d\{\ae\}monic ph\{\oe\}nix’s official r\"ole in fluffy souff\l\'es? }

\moreknutext
Text using only capital letters and some punctuation.

\newcommand{\@capknutext}{!
'THE DAZED BROWN FOX QUICKLY GAVE 12345---67890 JUMPS!')}
\let\capknutext\@capknutext

Lots of accents masquerading in personal names.
\def\@knunames{ \AA ngel\aa \ Beatrice Claire Diana \Eria Fran\c{c}oise Ginette H\'el\'ene Iris Jackie K\=aren \La\=ra Mar\'i\a \N\a\H{}a\ta{\l}a \u\i\e \O octave Pauline Qu\"eneau Roxanne Sabine T\~a\j\a \U r\v{s}ula Vivian Wendy Xanthippe Yv\{o}nne Z"azilie}

Just in case the French quotes are not defined, as they are called for in the subsequent \germantext.
\DeclareTextSymbol{\guillemotleft}{OT1}{'}\}
\DeclareTextSymbol{\guillemotright}{OT1}{'}\}
\providecommand{\flqq}{\guillemotleft}
\providecommand{\frqq}{\guillemotright}
\providecommand*{\germantext}{Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleich\-g\"ul\-lig ob ich schreibe: \frqq Dies ist ein Blindtext\flqq oder \frqq Huardest gefburn\flqq? Kjift -- mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und pr\"u\-fe, wie breit oder schmal sie \l"auft. Ein Blindtext sollte m\"og\-lichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss\ kein Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie \frqq Lorem ipsum\flqq dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.\par}

The traditional printers’ text.
\providecommand*{\latintext}{}%
\PackageWarning{fonttable}{\protect\latintext\space may be overridden by the babel package \MessageBreak use \protect\latinparatext\space instead}}
\providecommand{\latinparatext}{%
3.2 Testing a glyph

This is a reimplementation of Donald Knuth’s `testfont.tex` which is available from CTAN and there is also a commented version in Appendix H of *The METAFONT Book*.
\texttt{\texttt{\texttt{fnthours}}} \hspace{0.5cm} The time of day on a 24 hour clock.
\texttt{\texttt{\texttt{f@ttodigits}}} \hspace{0.5cm} The time of day on a 24 hour clock.

```
\newcommand*{\fnthours}{\@tempcntb=\time \divide\@tempcntb 60
\@tempcnta=-\@tempcntb \multiply\@tempcnta 60 \advance\@tempcnta \time
\texttt{\texttt{\texttt{f@ttodigits}}}@tempcnta}
\newcommand*{\f@ttodigits}{\ifnum #1<10 0\fi \number#1}
```

\texttt{\texttt{\texttt{\texttt{f@tgettsechars}}} (\texttt{T})\{\texttt{(S)}\}\{\texttt{(E)}\}) gets three characters and \texttt{chardefs} \texttt{f@ttchar} to \texttt{(T)} (the test character), \texttt{f@tschar} to \texttt{(S)} (start character) and \texttt{f@techar} to \texttt{(E)} (the end character).

```
\newcommand*{\f@tgettsechars}{[3]}{%
\chardef\f@ttchar=#1 \chardef\f@tschar=#2 \chardef\f@techar=#3}
```

\texttt{\texttt{\texttt{\texttt{glyphmixture}}} (\texttt{T})\{\texttt{(S)}\}\{\texttt{(E)}\}} sets a mix of \texttt{(T)} within the glyph range from \texttt{(S)} to \texttt{(E)} according to the pattern \texttt{f@tmixpattern}. The work is done by \texttt{\texttt{\texttt{\texttt{f@tdomix}}}}.

```
\newcommand*{\glyphmixture}{[3]}{%
\f@tdomix\f@tmixpattern}
```

\texttt{\texttt{\texttt{\texttt{\texttt{f@tmixpattern}}}}}

\texttt{\texttt{\texttt{\texttt{\texttt{f@taltpattern}}}}} These are similar to \texttt{\texttt{\texttt{\texttt{glyphmixture}}} and \texttt{\texttt{\texttt{\texttt{f@tmixpattern}}}} except that the glyphs are alternated.

```
\newcommand*{\glyphalternation}{[3]}{%
\f@tdomix\f@taltpattern}
```

\texttt{\texttt{\texttt{\texttt{f@tdisc}}} For breaking long lines so that the test character will be at the end of one line and repeated at the start of the next one.

```
\newcommand*{\f@tdisc}{\discretionary{\f@ttchar}{\f@ttchar}{\f@ttchar}}
```

\texttt{\texttt{\texttt{\texttt{f@tdoseries}}} (\texttt{T})\{\texttt{(S)}\}\{\texttt{(E)}\}} puts the test character \texttt{(T)} between all the others in the range \texttt{(S)} to \texttt{(E)}. The work is done by \texttt{\texttt{\texttt{\texttt{f@tdoseries}}}}.

```
\newcommand*{\glyphseries}{[3]}{%
\f@tdoseries\f@tschar\f@techar\par}
```

\texttt{\texttt{\texttt{\texttt{f@tdisc}}}}
\glyphalphabet\{⟨T⟩\} inserts the test glyph ⟨T⟩ between the lowercase alphabetic characters. Similarly \GLYPHALPHABET\{⟨T⟩\} does the same with the uppercase characters. The work is done by, respectively, \f@tcomplower and \f@tcompupper.

\newcommand*{\glyphalphabet}{\f@tcomplower}
\newcommand*{\GLYPHALPHABET}{\f@tcompupper}
\newcommand*{\f@tcomplower}[1]{\chardef\f@ttchar=#1
\f@tdisc\f@tdoseries{'a}{'z}\f@tdoseries{31}{34}\par}
\newcommand*{\f@tcompupper}[1]{\chardef\f@ttchar=#1
\f@tdisc\f@tdoseries{'A}{'Z}\f@tdoseries{35}{37}\par}
\newcommand*{\f@tdocomprehensive}{\par\chardef\f@ttchar=#2
\loop{#1} \ifnum\f@ttchar<#3\@tempcnta=\f@ttchar\advance\@tempcnta @\one \chardef\f@ttchar=\@tempcnta \repeat
\chardef\f@ttchar=#4
\loop{#1} \ifnum\f@ttchar<#5\@tempcnta=\f@ttchar\advance\@tempcnta @\one \chardef\f@ttchar=\@tempcnta \repeat}
\newcommand*{\f@tclc}{\f@tdisc\f@tdoseries{'a}{'z}\f@tdoseries{31}{34}\par}
\newcommand*{\f@tcuc}{\f@tdisc\f@tdoseries{'A}{'Z}\f@tdoseries{35}{37}\par}
\newcommand*{\f@tdgs}{\f@tdisc\f@tdoseries{'0}{'9}\par}
\newcommand*{\glyphpunct}{\par\f@tdopunct\{min\}\f@tdopunct\{pig\}\f@tdopunct\{hid\}\f@tdopunct\{HIE\}\f@tdopunct\{TIP\}\f@tdopunct\{fluff\}\$1,234.56 + 7/8 = 9\% @ \#0\par}
\newcommand*{\f@tdopunct}[1]{#1, #1: #1; '#1' '#1? '#1! (#1) [#1] #1* #1.\par}

The end of the package.

\section*{Bibliography}

\begin{enumerate}
\item Knut Lickert. \textit{Blindtext.sty: Creating text for testing / Texterzeugung zum testen}. October 2005. (Available from CTAN in macros/latex/contrib/blindtext)
\end{enumerate}
Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;</td>
<td>223, 234, 248, 252, 253</td>
</tr>
<tr>
<td>#</td>
<td>350</td>
</tr>
<tr>
<td>$</td>
<td>298, 350</td>
</tr>
<tr>
<td>%</td>
<td>350</td>
</tr>
<tr>
<td>&amp;</td>
<td>200, 298</td>
</tr>
<tr>
<td>-</td>
<td>248, 252, 253</td>
</tr>
<tr>
<td>=</td>
<td>232</td>
</tr>
<tr>
<td>@capknutext</td>
<td>226</td>
</tr>
<tr>
<td>@cclvi</td>
<td>90</td>
</tr>
<tr>
<td>@ehc</td>
<td>94</td>
</tr>
<tr>
<td>@firstoftwo</td>
<td>176</td>
</tr>
<tr>
<td>@knunames</td>
<td>230, 280</td>
</tr>
<tr>
<td>@moreknutext</td>
<td>222, 280</td>
</tr>
<tr>
<td>@tempa</td>
<td>19, 20, 30, 34</td>
</tr>
<tr>
<td>@tempb</td>
<td>20, 21</td>
</tr>
<tr>
<td>@tempcntb</td>
<td>300-303, 312-314, 324-326</td>
</tr>
<tr>
<td>\f</td>
<td>298</td>
</tr>
<tr>
<td>\f{}</td>
<td>298</td>
</tr>
<tr>
<td>\f^-</td>
<td>224, 233</td>
</tr>
<tr>
<td>\f^</td>
<td>298</td>
</tr>
<tr>
<td>\f^-'</td>
<td>223, 231, 237</td>
</tr>
<tr>
<td>\f^--'</td>
<td>233</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\AA</td>
<td>230</td>
</tr>
<tr>
<td>\aa</td>
<td>230</td>
</tr>
<tr>
<td>\AE</td>
<td>222</td>
</tr>
<tr>
<td>\ae</td>
<td>223</td>
</tr>
<tr>
<td>\A2text</td>
<td>295</td>
</tr>
<tr>
<td>\aztext</td>
<td>295</td>
</tr>
<tr>
<td>B</td>
<td>Numbers</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
</tr>
<tr>
<td>\baselineskip</td>
<td>50</td>
</tr>
<tr>
<td>C</td>
<td>Numbers</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
</tr>
<tr>
<td>\c</td>
<td>231</td>
</tr>
<tr>
<td>\capknutext</td>
<td>217, 226</td>
</tr>
<tr>
<td>\char</td>
<td>49, 58, 163, 325</td>
</tr>
<tr>
<td>\chardef</td>
<td>102, 307, 312, 313, 330, 332, 333, 340, 341, 343</td>
</tr>
<tr>
<td>D</td>
<td>Numbers</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
</tr>
<tr>
<td>\decimals</td>
<td>3, 175</td>
</tr>
<tr>
<td>\DeclareRobustCommand</td>
<td>12</td>
</tr>
<tr>
<td>\DeclareTextSymbol</td>
<td>236, 237</td>
</tr>
<tr>
<td>\digitstext</td>
<td>295</td>
</tr>
<tr>
<td>\discretionary</td>
<td>320</td>
</tr>
<tr>
<td>\dp</td>
<td>50, 167, 183, 184</td>
</tr>
<tr>
<td>E</td>
<td>Numbers</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
</tr>
<tr>
<td>\edef</td>
<td>18-20, 23, 30, 85, 91</td>
</tr>
<tr>
<td>\endgraf</td>
<td>313</td>
</tr>
<tr>
<td>F</td>
<td>Numbers</td>
</tr>
<tr>
<td>---</td>
<td>---------</td>
</tr>
<tr>
<td>\f@placechar</td>
<td>162, 176, 179, 181</td>
</tr>
<tr>
<td>\f@placedecimal</td>
<td>162, 179</td>
</tr>
</tbody>
</table>

Index


<table>
<thead>
<tr>
<th>S</th>
<th>T</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>\sevenrm</td>
<td>\tabskip</td>
<td>\v</td>
</tr>
<tr>
<td>\simpletext</td>
<td>\time</td>
<td>\vbox</td>
</tr>
<tr>
<td>\sixt@n</td>
<td>\tiny</td>
<td>\vrule</td>
</tr>
<tr>
<td>\smash</td>
<td>\ttfamily</td>
<td>X</td>
</tr>
<tr>
<td>\spaceskip</td>
<td>\u</td>
<td>\xdef</td>
</tr>
<tr>
<td>\ss</td>
<td>U</td>
<td>\xfonttable</td>
</tr>
<tr>
<td>\string</td>
<td>\u</td>
<td>\xspaceskip</td>
</tr>
</tbody>
</table>