The IFFONT package

Benjamin Weiss
weissbenjamin@me.com

Version 1.0.0 from 2015/11/11

1 Introduction

Using OpenType fonts within XeLaTeX or LuaTeX comes with a lot of benefits, but also the problem, that desired fonts may not be installed on the system, the user choses for the compilation. In case they are, there can still be different revisions of a font with variant names, such as *Frutiger LT Std*, *Frutiger Neue LT Pro* and *Frutiger Next Pro*.

Inspired by a question on Stack Exchange and a discussion on the XeTeX Mailinglist, this package addresses the problem with a set of easy to use macros: a macro to select the first font XeLaTeX or LuaTeX can find in a comma separated list and, additionally, a number of macro tests.

2 Select a Font from a List of Fonts

`\settofirstfound` This command defines a new macro with the first font found in the comma separated list. The list is provided as second argument. The macro can then be used to set the main font, set the sans font, define a new font family and so on. The following code shows how to set the main font as the Frutiger Font with the black font weight as bold font.

```
\documentclass{article}
\usepackage{fontspec}
\usepackage{iffont}
```

```
\settofirstfound{\mainfont}{Frutiger Next Pro,
                            Frutiger Neue LT Pro,
                            Frutiger LT Std}
\settofirstfound{\boldfont}{Frutiger Next Pro Black,
                            Frutiger Neue LT Pro Black,
                            Frutiger LT Std 75 Black}
\setmainfont[BoldFont=\boldfont]{\mainfont}
\begin{document}
\begin{itemize}
  \item \mainfont
  \item \textbf{\boldfont}
\end{itemize}
\end{document}
```

Defining a new font family works just the same.

```
\documentclass{article}
\usepackage{fontspec}
\usepackage{iffont}
\settofirstfound{\lightfont}{Frutiger Next Pro Light,
                             Frutiger Neue LT Pro Light,
                             Frutiger LT Std 45 Light}
\newfontfamily\FrutigerLight{\lightfont}
\begin{document}
\begin{itemize}
  \item {\FrutigerLight \lightfont}
\end{itemize}
\end{document}
```

3 Macro Tests

In most cases the `\settofirstfound` command is sufficient, but for the more demanding user there is also a set of macro tests. They are useful, if one wants to set specific font features only if the first choice font cannot be found. The original *Frutiger LT Std* font for example does not support small caps. So in the following example the `\textsc`command is redefined as uppercase with increased letter spacing, but only if the *Frutiger Next Pro* font cannot be found.

```
\documentclass{article}
\usepackage{fontspec}
\usepackage{iffont}
\settofirstfound{\mainfont}{Frutiger Next Pro,
                            Frutiger LT Std}
\setmainfont{\mainfont}
\ifxfontexists{Frutiger Next Pro}{%
  \def\textsc#1{\addfontfeature{LetterSpace=5.0}%
                \MakeUppercase{#1}}
}{}
\begin{document}
\begin{itemize}
  \item \mainfont
  \item \textsc{\mainfont}
\end{itemize}
\end{document}
```

4 Implementation

4.1 Required Packages

Load required packages.

```
\RequirePackage{fontspec}
\RequirePackage{etoolbox}
```

Init required variables.

```
\newtoggle{@iffont@fontfound}
\newcommand{\@iffont@firstfont}{Fira Sans}
\newcounter{@iffont@fontsnotfound}
```

`\iffontsexist` If all fonts are found the commands in the second argument will be executed, otherwise the commands in the third argument.

```
\newcommand{\iffontsexist}[3]{
  \setcounter{@iffont@fontsnotfound}{0}
  \expandafter\forcsvlist\expandafter\@iffont@checkfont\expandafter{#1}
```

```
  \ifnumequal{\value{@iffont@fontsnotfound}}{0}{%
    #2
  }{%
    #3
  }
}
```

`\ifxfontsexist` Same as `\iffontsexist`, but negated.

```
\newcommand{\ifxfontsexist}[3]{
  \iffontsexist{#1}{#3}{#2}
}
```

`\iffontexists` A simpler and therefore faster if clause, that only checks for a single font.

```
\newcommand{\iffontexists}[3]{
  \suppressfontnotfounderror=1
  \font\x = "#1" at 10pt
  \ifx\x\nullfont
    #3
  \else
    #2
  \fi
  \suppressfontnotfounderror=0
}
```

`\ifxfontexists` Same as `\iffontexists`, but negated.

```
\newcommand{\ifxfontexists}[3]{
  \iffontexists{#1}{#3}{#2}
}
```

`\settofirstfound` Sets the macro in the first argument to the first font in the comma separated list of fonts in the second argument that is found.

```
\newcommand{\settofirstfound}[2]{
  \togglefalse{@iffont@fontfound}
  \expandafter\forcsvlist\expandafter\@iffont@checkfont\expandafter{#2}
  \let#1\@iffont@firstfont
}
```

`\@iffont@checkfont` Checks if a font is found and increases the `@iffont@fontsnotfound` counter if not. The first font found will be saved in `\@iffont@firstfont`.

```
\newcommand{\@iffont@checkfont}[1]{
  \suppressfontnotfounderror=1
  \font\x = "#1" at 10pt
  \ifx\x\nullfont
    \stepcounter{@iffont@fontsnotfound}
  \else
    \nottoggle{@iffont@fontfound}{%
      \renewcommand{\@iffont@firstfont}{#1}
      \toggletrue{@iffont@fontfound}
    }{}
  \fi
  \suppressfontnotfounderror=0
}
```