

The IFFONT package

Benjamin Weiss

weissbenjamin@me.com

Version 1.0.0 from 2015/11/08

1 Introduction

The power of using OpenType fonts within X_YTeX or LuaTeX comes with the problem, that a certain font may not be available on every system one might want to compile a certain document. Or there are different revisions of a font with different names like *Frutiger LT Std*, *Frutiger Neue LT Pro* and *Frutiger Next Pro*.

Inspired by a question on [Stack Exchange](#) and a discussion on the [X_YTeX Mailinglist](#) this package addresses this problem with a set of easy to use macros: A macro to select the first font in a comma separated list X_YTeX or LuaTeX can find and a number of macro tests.

2 Select Font from a List of Fonts

`\settofirstfound` This command defines a new macro with the first font found in the comma separated list provided as second argument. This macro can then for example be used to set the main font. If we consider the Frutiger example from the introduction the main font with the black font weight as bold font can be set like this.

```
\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}

```

The same works naturally for defining a new font family.

```

\documentclass{article}
\usepackage{fontspec}
\usepackage{iffont}
\settofirstfound{\lightfont}{Frutiger Next Pro Light,
Frutiger Neue LT Pro Light,
Frutiger LT Std 45 Light}
\newfontfamily\Light{Fira Sans Light}
\begin{document}
\begin{itemize}
\item {\Light \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. These are for example useful if one wants to set specific font features only if the first choice font cannot be found. Let us again consider our Frutiger example. The original *Frutiger LT Std* font 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.

```

1 \RequirePackage{fontspec}
2 \RequirePackage{etoolbox}

```

Init required variables.

```

3 \newtoggle{@iffont@fontfound}
4 \newcommand{\@iffont@firstfont}{Fira Sans}
5 \newcounter{@iffont@fontnotfound}

```

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

```

6 \newcommand{\iffontsexist}[3]{
7   \setcounter{@iffont@fontnotfound}{0}
8   \expandafter\forcsvlist\expandafter\@iffont@checkfont\expandafter{#1}

```

```

9 \ifnumequal{\value{@iffont@fontsnofound}}{0}{%
10     #2
11 }{%
12     #3
13 }
14 }

```

\iffontsexist Same as **\iffontsexist** but negated.

```

15 \newcommand{\iffontsexist}[3]{
16     \iffontsexist{#1}{#3}{#2}
17 }

```

\iffontexists A simpler and therefore faster if clause that only checks if one font is found.

```

18 \newcommand{\iffontexists}[3]{
19     \suppressfontnotfounderror=1
20     \font\x = "#1" at 10pt
21     \ifx\x\nullfont
22         #3
23     \else
24         #2
25     \fi
26     \suppressfontnotfounderror=0
27 }

```

\ifxfontexists Same as **\iffontexists** but negated.

```

28 \newcommand{\ifxfontexists}[3]{
29     \iffontexists{#1}{#3}{#2}
30 }

```

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

```

31 \newcommand{\settofirstfound}[2]{
32     \togglefalse{@iffont@fontfound}
33     \expandafter\forcsvlist\expandafter{@iffont@checkfont}\expandafter{#2}
34     \let#1\@iffont@firstfont
35 }

```

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

```
36 \newcommand{\@iffont@checkfont}[1]{
37   \suppressfontnotfounderror=1
38   \font\x = "#1" at 10pt
39   \ifx\x\nullfont
40     \stepcounter{@iffont@fontsnofound}
41   \else
42     \nottoggle{@iffont@fontfound}{%
43       \renewcommand{\@iffont@firstfont}{#1}
44       \toggletrue{@iffont@fontfound}
45     }{}
46   \fi
47   \suppressfontnotfounderror=0
48 }
```