`ffcode`: LaTeX Package for Fixed-Font Code Blocks*

Yegor Bugayenko
yegor256@gmail.com

2022-12-02, 0.8.0

1 Introduction

This package helps you write source code in your articles and make sure it looks nice. Install it from CTAN and then use like this (pay attention to `\ff` command and `ffcode` environment):

The function `fibo()` is recursive:

```
int fibo(int n) {
  if (n < 2) {
    return n;
  }
  return fibo(n-1)+fibo(n-2);
}
```

Line no. 3 returns `n` and terminates it.

```
\documentclass{article}
\usepackage{ffcode}
\pagestyle{empty}
\begin{document}
The function |fibo()| is recursive:
\begin{ffcode}
int fibo(int n) {
  if (n < 2) {
    return n; |\label{ln:ret}|
  }
  return fibo(n-1)+fibo(n-2);
}
\end{ffcode}
Line no.~\ref{ln:ret} returns \ff{n}
and terminates it.
\end{document}
```

2 Package Options

`nopygments` You have to run `pdflatex` with `-shell-escape` flag in order to let `minted` (the package we use) to run Pygments and format the code. If you don't want this to happen, just use `nopygments` option.

`noframes` If you want to omit the light gray frames around `\ff` texts, use the package option `noframes`.

`nobars` To omit the vertical gray bar at the left side of each snippet, use `nobars` option of the package.

*The sources are in GitHub at yegor256/ffcode

`nonumbers` To omit the line numbers, use `nonumbers` option of the package.

`nocn` By default, the numbering is continuous: line numbers start at the first snippet and increment until the end of the document. If you want them to start from one at each snippet, use `nocn` (stands for "no continuous numbering") option of the package.

`bold` You can make your `\ff` pieces look bolder than usual, which may be pretty convenient for some document classes (pay attention to the usage of the lmodern package, without it the bold won't work, as explained here):

Sometimes it's necessary to make code pieces look bolder, like the `fibo()` function in this text.

```
\usepackage{lmodern}
\usepackage[bold,noframes]{ffcode}
\begin{document}
Sometimes it's necessary to make
code pieces look bolder, like
the |fibo()| function in this text.
\end{document}
```

`sf` You can change the font family of `\ff` pieces to `\sffamily`:

Sometimes you may want them to look not strictly fixed-width, but more elegant, like the **fibo()** here.

```
\usepackage[sf,bold,noframes]{ffcode}
\begin{document}
Sometimes you may want them to look
not strictly fixed-width, but more
elegant, like the \emph{|fibo()|}
here.
\end{document}
```

3 Typesetting

By the way, the package correctly formats low-height texts, for example, just a dot: `.`

A pair of vertical lines decorate a TeX command inside the snippet. If you want to print a single vertical line, use this: "`|\char‘\\vert|`".

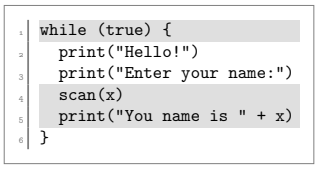
The command `\ff` behaves differently in math mode — it doesn't add gray frames:

$$x = \int_{\text{home}}^{N} f(x).$$

```
\begin{equation*}
x = \int_{|home|}^N f(x).
\end{equation*}
```

4 Line Highlighting

You can highlight some lines in your `ffcode` environment, or can use any other additional configuration parameters from `minted` package:



```
while (true) {
  print("Hello!")
  print("Enter your name:")
  scan(x)
  print("You name is " + x)
}
```

```
\begin{ffcode*}{highlightlines={1,4-5}}
while (true) {
  print("Hello!")
  print("Enter your name:")
  scan(x)
  print("You name is " + x)
}
\end{ffcode*}
```

Using this second argument of `ffcode*` (with the trailing asterisk), you can provide any other options from `minted` package to the snippet.

5 Implementation

First, we parse package options with the help of pgfopts package:

```
\RequirePackage{pgfopts}
\pgfkeys{
  /ff/.cd,
  bold/.store in=\ff@bold,
  sf/.store in=\ff@sf,
  nopygments/.store in=\ff@nopygments,
  noframes/.store in=\ff@noframes,
  nonumbers/.store in=\ff@nonumbers,
  nobars/.store in=\ff@nobars,
  novert/.store in=\ff@novert,
  nocn/.store in=\ff@nocn,
}
\ProcessPgfPackageOptions{/ff}
```

Then, we disable pygments for minted, if necessary:

```
\makeatletter\ifdefined\ff@nopygments
  \PassOptionsToPackage{draft=true}{minted}
\fi\makeatother
```

Then, we configure minted package:

```
\RequirePackage{minted}
\setminted{breaklines}
\setminted{escapeinside=||,mathescape}
\setminted{highlightcolor=gray!25}
\usemintedstyle{bw}
```

`ffcode` Then, we define `ffcode` environment:

```
\makeatletter\ifdefined\ff@nonumbers
  \ifdefined\ff@nobars
    \newminted[ffcode]{text}{}
  \else
    \newminted[ffcode]{text}{framesep=6pt,
      framerule=1pt,rulecolor=gray,frame=leftline}
  \fi
\else
  \renewcommand{\theFancyVerbLine}{\textcolor{gray}%
    {\tiny\oldstylenums{\ttfamily\arabic{FancyVerbLine}}}}
```

```
\ifdefined\ff@nocn
  \ifdefined\ff@nobars
    \newminted[ffcode]{text}{
      linenos,numbersep=2pt
    }
  \else
    \newminted[ffcode]{text}{
      framesep=6pt,framerule=1pt,rulecolor=gray,
      frame=leftline,linenos,numbersep=2pt
    }
  \fi
\else
  \ifdefined\ff@nobars
    \newminted[ffcode]{text}{
      firstnumber=last,linenos,numbersep=2pt
    }
  \else
    \newminted[ffcode]{text}{
      framesep=6pt,framerule=1pt,rulecolor=gray,
      frame=leftline,firstnumber=last,linenos,numbersep=2pt
    }
  \fi
\fi\makeatother
```

`\ff@print` Then, we define a supplementary macro `\ff@print`:

```
\makeatletter
\newcommand\ff@print[1]{%
  \textnormal{%
    \ifdefined\ff@sf\sffamily\else\ttfamily\fi%
    \ifdefined\ff@bold\fontseries{b}\selectfont\fi%
    #1%
  }%
}
\makeatother
```

`\ff@rule` Then, we define supplementary command `\ff@rule`:

```
\makeatletter\newcommand\ff@rule
  {\vrule height 6pt depth 1pt width 0pt}
\makeatother
```

`tcolorbox` Then, we use tcolorbox to define `\ff@box` command for a gray box around verbatim text block:

```
\makeatletter
\ifdefined\ff@noframes\else
  \RequirePackage{tcolorbox}
  \newtcbox\ff@box{nobeforeafter,colframe=gray!80!white,
    colback=gray!5!white,boxrule=0.1pt,arc=1pt,
    boxsep=1.2pt,left=0.5pt,right=0.5pt,top=0.2pt,bottom=0.2pt,
    tcbox raise base}
\fi
\makeatother
```

`\ff@x` Then, we define `\ff@x` internal command for printing a piece of fixed-width-font text:

```
\makeatletter
\NewDocumentCommand\ff@x{v}{\ff{#1}}
\makeatother
```

`\ff` Then, we define `\ff` macro:

```
\makeatletter
\newcommand\ff[1]{%
  \ifdefined\ff@noframes%
    \ff@rule\ff@print{#1}%
  \else%
    \relax\ifmmode%
      \ff@rule\ff@print{#1}%
    \else%
      \ff@box{\ff@rule\ff@print{#1}}%
    \fi%
  \fi%
}
\makeatother
```

`novert` Finally, we let vertical bars work similar to `\ff`, as suggested here and here (unless `novert` package option is used):

```
\makeatletter\ifdefined\ff@novert\else
  \catcode`\|\active
  \AtBeginDocument{\catcode`\|\active\protected\def|{\ff@x|}}
  \catcode`\| 12 %
\fi\makeatother
```

Change History

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.