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

Yegor Bugayenko

yegor256@gmail.com

2025/07/06, 0.10.2

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):

<div>

The function `fibo()` is recursive:

```
1 int fibo(int n) {
2   if (n < 2) {
3     return n;
4   }
5   return fibo(n-1)+fibo(n-2);
6 }
```

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

</div>

```
1  \documentclass{article}
2  \usepackage{ffcode}
3  \pagestyle{empty}
4  \begin{document}
5  The function |fibo()| is recursive:
6  \begin{ffcode}
7  int fibo(int n) {
8    if (n < 2) {
9      return n; (*@ \label{ln:ret} @*)
10   }
11   return fibo(n-1)+fibo(n-2);
12 }
13 \end{ffcode}
14 Line no.~\ref{ln:ret} returns \ff{n}
15 and terminates it.
16 \end{document}
```

`\ffinput`   There is also an `\ffinput` command that reads the content from a file and places it into the document, formatted exactly the same way as it would be if the `ffcode` environment were used.

2 Package Options

`samepage`   If you want to avoid page breaks inside `ffcode` blocks, use the `samepage` package option.

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

---

[*]The sources are in GitHub at [yegor256/ffcode](https://github.com/yegor256/ffcode)

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

nonumbers To omit the line numbers, use the `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.

```
4  \usepackage{lmodern}
5  \usepackage[bold,noframes]{ffcode}
6  \begin{document}
7  Sometimes it's necessary to make
8  code pieces look bolder, like
9  the |fibo()| function in this text.
10 \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.

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

3   Typesetting

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

A pair of vertical lines decorates 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:

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

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

4   Line Highlighting

You can highlight some lines in your `ffcode` environment, or can use any other additional configuration parameters from `fancyvrb` and `fvextra` packages:

```
1  while (true) {
2    print("Hi!")
3    print("Enter your name:")
4    scan(x)
5    print("Your name | " + x)
6  }
```

```
6  \begin{ffcode}[backgroundcolor=\color{gray!20}]
7  while (true) {
8     (*@\textcolor{red}{print("Hi!")}@*)
9     print("Enter your name:")
10    scan(x)
11    print("Your name | " + x)
12 }
13 \end{ffcode}
```

Using this second argument of `ffcode`, you can provide any other options from the `listings` package to the snippet.

5 Implementation

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

```
1  \RequirePackage{pgfopts}
2  \pgfkeys{
3    /ff/.cd,
4    bold/.store in=\ff@bold,
5    sf/.store in=\ff@sf,
6    samepage/.store in=\ff@samepage,
7    noframes/.store in=\ff@noframes,
8    nonumbers/.store in=\ff@nonumbers,
9    nobars/.store in=\ff@nobars,
10   novert/.store in=\ff@novert,
11   nocn/.store in=\ff@nocn,
12 }
13 \ProcessPgfPackageOptions{/ff}
```

Then, we include the xcolor package:

```
14 \RequirePackage{xcolor}
```

Then, we configure the listings package:

```
15 \RequirePackage{listings}
16 \makeatletter
17 % See \href{https://tex.stackexchange.com/questions/706858}{the explanation}:
18 \lst@AddToHook{Init}{\setlength{\lineskip}{0pt}}
19 \makeatother
```

Then, we define a supplementary command \ff@set:

```
20 \makeatletter
21 \newcommand\ff@set[1]{
22   \lstset{breaklines}
23   \lstset{escapeinside={(*@}{@*)}}
24   \lstset{basicstyle={\ttfamily}}
25   \lstset{columns=fullflexible}
26   \lstset{keepspaces=true}
27   \ifdefined\ff@nonumbers\else
28     \lstset{numbers=left,numbersep=.8em,numberstyle={\tiny\sffamily\color{gray}}}
29   \fi
30   \ifdefined\ff@nobars\else
31     \lstset{frame=leftline,framerule=.05em,rulecolor={\color{gray}}}
32   \fi
```

```
33    \ifdefined\ff@nocn\else
34      \lstset{firstnumber=last}
35    \fi
36    \lstset{#1}
37 }
38 \makeatother
```

ffcode Then, we define the `ffcode` environment and its supplementary `ffcode*` counterpart:

```
39 \makeatletter
40 \lstnewenvironment{ffcode}[1][]
41   {\ff@set{#1}\ifdefined\ff@samepage\noindent\minipage{\linewidth}\fi}
42   {\ifdefined\ff@samepage\endminipage\fi}
43 \makeatother
```

ffcode Then, we define the `\ffinput` command:

```
44 \makeatletter
45 \newcommand\ffinput[2][]{%
46   \ff@set{#1}%
47   \ifdefined\ff@samepage\noindent\minipage{\linewidth}\fi%
48   \lstinputlisting[#1]{#2}%
49   \ifdefined\ff@samepage\endminipage\fi%
50 }
51 \makeatother
```

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

```
52 \makeatletter
53 \newcommand\ff@print[1]{%
54   \textnormal{%
55     \ifdefined\ff@sf\sffamily\else\ttfamily\fi%
56     \ifdefined\ff@bold\fontseries{b}\selectfont\fi%
57     #1%
58   }%
59 }
60 \makeatother
```

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

```
61 \makeatletter\newcommand\ff@rule
62   {\vrule height 0.6em depth 0.1em width 0em}
63 \makeatother
```

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

```
64 \makeatletter
65 \ifdefined\ff@noframes\else
66   \RequirePackage{tcolorbox}
67   \newtcbox\ff@box{nobeforeafter,colframe=gray!80!white,
68     colback=gray!5!white,boxrule=0.01em,arc=0.1em,
69     boxsep=0.12em,left=0.05em,right=0.05em,top=0.02em,bottom=0.02em,
70     tcbox raise base}
71 \fi
72 \makeatother
```

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

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

\ff Then, we define \ff macro:

```
76 \makeatletter
77 \newcommand\ff[1]{%
78   \ifdefined\ff@noframes%
79     \ff@rule\ff@print{#1}%
80   \else%
81     \relax\ifmmode%
82       \ff@rule\ff@print{#1}%
83     \else%
84       \ff@box{\ff@rule\ff@print{#1}}%
85     \fi%
86   \fi%
87 }
88 \makeatother
```

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

```
89 \makeatletter\ifdefined\ff@novert\else
90   \catcode'\|\active
91   \AtBeginDocument{\catcode'\|\active\protected\def|{\ff@x|}}
92   \catcode'\| 12 %
93 \fi\makeatother
```

```
94 \endinput
```

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.