

`ffcode`: L^AT_EX Package for Fixed-Font Code Blocks*

Yegor Bugayenko
yegor256@gmail.com

2026-10-14, 0.13.1

1 Introduction

This package typesets source code within articles and ensures a tidy, consistent appearance. Following installation from CTAN, the typical usage proceeds as follows (note the `\ff` command and the `ffcode` environment):

The function <code>fibo()</code> is recursive: <pre>1 int fibo(int n) { 2 if (n < 2) { 3 return n; 4 } 5 return fibo(n-1)+fibo(n-2); 6 }</pre> Line no. 3 returns <code>n</code> and terminates it.	<pre>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}</pre>
--	--

`\ffinput` An `\ffinput` command is also provided, which reads content from a file and inserts it into the document, formatted in precisely the manner that would result from the `ffcode` environment. The `ffsave` environment may be employed beforehand; it behaves as the `ffcode` environment, save that the content is written to a file rather than printed, whence it may subsequently be retrieved by the `\ffinput` command. The name of that file is given in the optional argument of `ffsave`, for instance `\begin{ffsave}[foo.txt]`, and defaults to `ffsave.txt`.

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

2 Package Options

Each option below is a switch: naming it turns it on, and it also accepts an explicit value, either `true` or `false`. For instance, `\usepackage[nonumbers=false]{ffcode}` keeps the line numbers.

- samepage** The `samepage` package option prevents page breaks inside `ffcode` blocks.
- noframes** The `noframes` package option omits the light gray frames around `\ff` texts.
- nobars** The `nobars` package option omits the vertical gray bar at the left side of each snippet.
- nonumbers** The `nonumbers` package option omits the line numbers.
- novert** By default, a pair of vertical bars works as a shortcut for `\ff`. The `novert` package option leaves the vertical bar as an ordinary character.
- nocn** By default, line numbering is continuous: numbers commence at the first snippet and increment until the end of the document. The `nocn` package option (an abbreviation for “no continuous numbering”) causes the numbering to restart from one in each snippet.
- tmpfile** Every `ffcode` block is captured verbatim to a scratch file before it is typeset (this is what makes `\ffcolumbreak` possible). The name of that file defaults to `\jobname.ffcode` and may be changed with the `tmpfile` package option, for instance `\usepackage[tmpfile=build/ffcode.tmp]{ffcode}`.
- bold** The `\ff` fragments may be rendered in a heavier weight than the default, which proves convenient for certain document classes (note the use of the [lmodern](#) package: without it, boldface fails to operate, as explained [here](#)):

Sometimes it's necessary to make code pieces look bolder, like the `fibonacci` 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 |fibonacci| function in this text.
10 \end{document}

```

- sf** The font family of `\ff` fragments may be altered to `\sffamily`:

Sometimes a less strictly fixed-width, more elegant look may be preferred, like the `fibonacci` here.

```

4 \usepackage[sf,bold,noframes]{ffcode}
5 \begin{document}
6 Sometimes a less strictly fixed-width,
7 more elegant look may be preferred,
8 like the \emph{|fibonacci|} here.
9 \end{document}

```

3 Typesetting

The package handles low-height text correctly, for instance a solitary dot: `.`

A pair of vertical bars decorates a TeX command within a snippet. A single vertical bar may be produced by means of `\ff{\char‘\}`.

The `\ff` command behaves differently in math mode—it adds no gray frames:

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

```

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

```

Lines within the `ffcode` environment may be highlighted, and any further configuration parameters from the `listings` package may likewise be supplied:

```

1 while (true) {
2   print("Hi!")
3   print("Enter your
4     name:")
5   scan(x)
6   print("Your name | "
7     + x)
8 }

```

```

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}

```

`\ffcolumnbreak` A snippet may be split into two or more side-by-side columns by placing `\ffcolumnbreak` on a line of its own, wrapped in an escape so that `listings` does not print it. Line numbering runs continuously across the columns, which is handy for a listing too tall for the page. Each further `\ffcolumnbreak` adds one more column. By default the columns share the line equally; a column whose longest line does not fit in its equal share is widened just enough to hold it, and the extra space is taken from the columns that have room to spare:

```

1 int fibo(
2   int n
3 ) {
4   if (n <
5     2) {
6
7     return n;
8   }
9   return fibo(n
10    -1)+fibo(n-2)
11 ;
12 }

```

```

6 \begin{ffcode}
7 int fibo(int n) {
8   if (n < 2) {
9     (*\ffcolumnbreak @*)
10    return n;
11   }
12   return fibo(n-1)+fibo(n-2);
13 }
14 \end{ffcode}

```

The `\lstset` command of the `listings` package may also be employed to establish defaults for the `ffcode` environment. In contrast to the optional argument, which applies solely to a single snippet, `\lstset` acts globally: every subsequent code listing in the document inherits the new settings, including any further `ffcode` blocks thereafter. To confine the change to a single `ffcode` block, the call should be wrapped within a `\begin{group}... \end{group}` pair:

```

1 print("Hi!")

```

```

6 \begin{group}
7 \lstset{backgroundcolor=\color{yellow!30}}
8 \begin{ffcode}
9 print("Hi!")
10 \end{ffcode}
11 \end{group}

```

A snippet may be rendered in “condensed” mode, with reduced spacing between letters, by supplying `columns=fixed` together with a narrower `basewidth` from

the `listings` package:

<pre>1 while (true) { 2 print("Enter your name:") 3 scan(x) 4 print("Hello, " + x) 5 }</pre>	<pre>6 \begin{ffcode}[columns=fixed,basewidth=0.4em] 7 while (true) { 8 print("Enter your name:") 9 scan(x) 10 print("Hello, " + x) 11 } 12 \end{ffcode}</pre>
--	---

4 Implementation

First, we parse package options with the help of [pgfopts](#) package:

```
1 \RequirePackage{pgfopts}  
2 \newif\iff@bold  
3 \newif\iff@sf  
4 \newif\iff@samepage  
5 \newif\iff@noframes  
6 \newif\iff@nonumbers  
7 \newif\iff@nobars  
8 \newif\iff@novert  
9 \newif\iff@nocn  
10 \pgfkeys{  
11   /ff/.cd,  
12   bold/.is if=ff@bold,  
13   sf/.is if=ff@sf,  
14   samepage/.is if=ff@samepage,  
15   noframes/.is if=ff@noframes,  
16   nonumbers/.is if=ff@nonumbers,  
17   nobars/.is if=ff@nobars,  
18   novert/.is if=ff@novert,  
19   nocn/.is if=ff@nocn,  
20   tmpfile/.store in=\ff@file,  
21   tmpfile/.default=\jobname.ffcode,  
22   tmpfile,  
23 }  
24 \ProcessPgfPackageOptions{/ff}
```

Then, we include the [xcolor](#) package:

```
25 \RequirePackage{xcolor}
```

Then, we configure the [listings](#) package. The defaults are applied here, at package load, so that a user may override them globally with `\lstset` after `\usepackage{ffcode}`. The line skip is reset to zero at the start of every listing, as [explained here](#). Also, `acmart` warns about the `\@vspace` that `listings` emits around displays, so its saved originals are restored during typesetting:

```
26 \RequirePackage{listings}  
27 \lst@AddToHook{Init}{\setlength{\lineskip}{0pt}}  
28 \ifdefined\@vspace@orig  
29   \lst@AddToHook{DisplayStyle}{%  
30     \let\@vspace\@vspace@orig  
31     \ifdefined\@vspacer@orig\let\@vspacer\@vspacer@orig\fi}
```

```

32 \fi
33 \lstset{breaklines}
34 \lstset{escapeinside={(*@){@*}}}
35 \lstset{basicstyle={\ttfamily}}
36 \lstset{columns=fullflexible}
37 \lstset{keepspaces=true}
38 \iff@nonumbers\else
39   \lstset{numbers=left,numbersep=.8em,
40     numberstyle={\tiny\sffamily\color{gray}},
41     xleftmargin=2.5em}
42 \fi
43 \iff@nobars\else
44   \lstset{frame=leftline,framerule=.05em,rulecolor={\color{gray}}}
45 \fi
46 \iff@nocn\else
47   \lstset{firstnumber=last}
48 \fi

```

`\ffcolumnbreak` Then, we define the `\ffcolumnbreak` marker and the machinery that splits an `ffcode` block into side-by-side columns. Because `listings` typesets a single line-by-line stream, a block is first captured verbatim to a temporary file; that file is then scanned for `\ffcolumnbreak` markers, which cut it into segments. Only a line that holds nothing but the escaped marker, spaces aside, counts as a break; any other line that mentions `\ffcolumnbreak` is left to `listings`. Each segment is measured before it is printed: it is typeset once into a discarded box, so that `listings` records the width of its longest line in `\lst@maxwidth`, and that width (plus the gutter reserved for line numbers) becomes the natural width of the column. The available width is then shared by a water-filling pass in `\ff@computelevel`: it seeks the common level such that every column narrower than the level takes the level, while a column wider than the level keeps its natural width, and the two together fill `\linewidth`. Thus equal-sized columns split the line evenly, and only a column too wide for its share claims more, at the expense of the columns that have room to spare. When even the natural widths overflow `\linewidth`, the widths are scaled down in proportion to fit. A block without any marker is rendered as one listing, exactly as before:

```

49 \RequirePackage{fancyvrb}
50 \newcommand\ffcolumnbreak{}
51 \def\ff@flat#1{\zap@space#1 \@empty}
52 \edef\ff@cbmarker{\detokenize{(*@)\string\ffcolumnbreak\detokenize{(*)}}}
53 \newread\ff@read
54 \newcount\ff@nlines
55 \newcount\ff@nbreaks
56 \newcount\ff@ncols
57 \newcount\ff@prev
58 \newcount\ff@start
59 \newdimen\ff@colwd
60 \newdimen\ff@sepwd
61 \newdimen\ff@total
62 \newdimen\ff@avail
63 \newdimen\ff@natwd
64 \newdimen\ff@gutter
65 \newdimen\ff@level
66 \newdimen\ff@fixedsum

```

```

67 \newcount\ff@fixedcnt
68 \newcount\ff@denom
69 \newcount\ff@prevcnt
70 \newsavebox\ff@mbox
71 \let\ff@grab\relax
72 \lst@AddToHook{Init}{\ff@grab}
73 \def\ff@colsep{\hspace{0.4em}{\color{gray}\vrule width 0.3pt}}%
74 \hspace{0.4em}}
75 \def\ff@allother{%
76   \count@=\z@
77   \loop\catcode\count@=12 \ifnum\count@<255 \advance\count@\@ne\repeat
78   \catcode'\ =10 \endlinechar=-1 }
79 \def\ff@readloop{%
80   \read\ff@read to\ff@line
81   \ifeof\ff@read
82     \let\ff@next\relax
83   \else
84     \global\advance\ff@nlines\@ne
85     \edef\ff@flatline{\expandafter\ff@flat\expandafter{\ff@line}}%
86     \ifx\ff@flatline\ff@cbmarker
87       \global\advance\ff@nbreaks\@ne
88       \xdef\ff@segs{\unexpanded\expandafter{\ff@segs}\noexpand\ff@seg
89         {\the\numexpr\ff@start+\@ne\relax}{\the\numexpr\ff@nlines-\@ne\relax}}%
90       \global\ff@start=\ff@nlines
91     \fi
92     \let\ff@next\ff@readloop
93   \fi
94   \ff@next}
95 \def\ff@scan{%
96   \global\ff@nlines=\z@ \global\ff@nbreaks=\z@
97   \global\ff@start=\z@ \gdef\ff@segs{}%
98   \openin\ff@read=\ff@file\relax
99   \begingroup\ff@allother\ff@readloop\endgroup
100  \closein\ff@read
101  \ifnum\ff@nbreaks>\z@
102    \xdef\ff@segs{\unexpanded\expandafter{\ff@segs}\noexpand\ff@seg
103      {\the\numexpr\ff@start+\@ne\relax}{\the\ff@nlines}}%
104  \fi}
105 \def\ff@measure#1#2{%
106   \setbox\ff@mbox=\vbox{\hfuzz\maxdimen
107     \lstinputlisting[numbers=none,breaklines=false,%
108       firstline=#1,lastline=#2]{\ff@file}}%
109   \ff@natwd=\dimexpr\lst@maxwidth+\ff@gutter\relax
110   \global\advance\ff@total\ff@natwd
111   \xdef\ff@cols{\unexpanded\expandafter{\ff@cols}}%
112   \noexpand\ff@col{#1}{#2}{\the\ff@natwd}}
113 \def\ff@onepass{%
114   \global\ff@fixedsum=\z@ \global\ff@fixedcnt=\z@
115   \begingroup
116     \def\ff@col##1##2##3{%
117       \ifdim##3>\ff@level
118         \global\advance\ff@fixedsum##3\relax
119         \global\advance\ff@fixedcnt\@ne
120       \fi}%

```

```

121 \ff@cols
122 \endgroup
123 \ifnum\ff@fixedcnt<\ff@ncols
124 \ff@denom=\numexpr\ff@ncols-\ff@fixedcnt\relax
125 \ff@level=\dimexpr(\ff@avail-\ff@fixedsum)/\ff@denom\relax
126 \fi}
127 \def\ff@computelevel{%
128 \ff@level=\dimexpr\ff@avail/\ff@ncols\relax
129 \ff@prevcnt=\z@
130 \loop
131 \ff@onepass
132 \ifnum\ff@fixedcnt>\ff@prevcnt
133 \ff@prevcnt=\ff@fixedcnt
134 \repeat}
135 \def\ff@col#1#2#3{%
136 \ifnum\ff@prev=\z@ \else \ff@colsep\lstset{firstnumber=last}\fi
137 \ifdim\ff@total>\ff@avail
138 \ff@colwd=\dimexpr#3*\number\ff@avail/\number\ff@total\relax
139 \else
140 \ifdim#3>\ff@level\ff@colwd=#3\relax \else \ff@colwd=\ff@level\fi
141 \fi
142 \parbox[t]{\ff@colwd}{%
143 \ifdefined\@vspace@orig
144 \let\@vspace\@vspace@orig
145 \ifdefined\@vspacer@orig \let\@vspacer\@vspacer@orig\fi
146 \fi
147 \vspace{0pt}%
148 \lstinputlisting[firstline=#1,lastline=#2]{\ff@file}}%
149 \ff@prev=\@ne}
150 \def\ff@render{%
151 \ff@scan
152 \begingroup
153 \expandafter\lstset\expandafter{\ff@opts}%
154 \ifnum\ff@nbreaks=\z@
155 \lstinputlisting{\ff@file}%
156 \else
157 \ff@ncols=\ff@nbreaks \advance\ff@ncols\@ne
158 \ff@sepwd=0.8em \advance\ff@sepwd0.3pt
159 \ff@avail=\dimexpr\linewidth-\ff@sepwd*\ff@ncols+\ff@sepwd\relax
160 \global\ff@total=\z@ \gdef\ff@cols{%
161 \edef\ff@savenum{\the\c@lstnumber}%
162 \def\ff@grab{\global\ff@gutter\@totalleftmargin}%
163 \let\ff@seg\ff@measure \ff@segs
164 \let\ff@grab\relax
165 \global\c@lstnumber=\ff@savenum\relax
166 \ff@computelevel
167 \par\addvspace\topsep
168 \noindent\hbox{\ff@prev=\z@ \ff@cols}%
169 \par\addvspace\topsep
170 \fi
171 \endgroup}

```

`ffcode` (*env.*) Then, we (re)define the `ffcode` environment. Its body is written verbatim to the temporary file and handed to `\ff@render`. The optional [...] argument is

parsed the same way `fancyvrb` does it, by turning the end-of-line active before looking ahead, so that the newline after `\begin{ffcode}` is not swallowed when no options are supplied:

```

172 \newenvironment{ffcode}
173   {\catcode'\^^M=\active
174    \@ifnextchar[\ff@begin{\ff@begin[]}}
175   {\end{VerbatimOut}}%
176    \ff@render
177    \iff@samepage\endminipage\fi}
178 \def\ff@begin[#1]{%
179   \catcode'\^^M=5\relax
180   \def\ff@opts{#1}%
181   \iff@samepage\noindent\minipage{\linewidth}\fi
182   \VerbatimEnvironment\begin{VerbatimOut}{\ff@file}}

```

`\ffinput` Then, we define the `\ffinput` command. It hands its file and options to `\ff@render`, inside a group, so that it prints a file exactly as the `ffcode` environment would, columns included:

```

183 \newcommand\ffinput[2][]{%
184   \iff@samepage\noindent\minipage{\linewidth}\fi%
185   \begingroup\def\ff@file{#2}\def\ff@opts{#1}\ff@render\endgroup
186   \iff@samepage\endminipage\fi%
187 }

```

`ffsave (env.)` Then, we define the `ffsave` environment. Before opening the output stream, its missing parent directory is created, so that `\begin{ffsave}[foo/bar.txt]` works even when `foo/` is absent (this needs shell escape enabled):

```

188 \RequirePackage{shelless}
189 \ExplSyntaxOn
190 \cs_new_protected:cpn { ff@mkdir } #1
191   {
192     \str_set:Nn \l_tmpa_str {#1}
193     \sys_if_platform_windows:TF
194       {
195         \str_replace_all:Nnn \l_tmpa_str { / } { \c_backslash_str }
196         \ShellEscape { if~not~exist~"\l_tmpa_str"~mkdir~"\l_tmpa_str" }
197       }
198       { \ShellEscape { mkdir~-p~"\l_tmpa_str" } }
199   }
200 \ExplSyntaxOff
201 \newenvironment{ffsave}[1][ffsave.txt]
202   {\filename@parse{#1}%
203    \ifx\filename@area\@empty\else
204      \expandafter\ff@mkdir\expandafter{\filename@area}%
205    \fi
206    \VerbatimEnvironment\begin{VerbatimOut}{#1}}
207   {\end{VerbatimOut}}

```

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

```

208 \newcommand\ff@print[1]{%
209   \textnormal{%
210     \iff@sffamily\else\ttfamily\fi%
211     \iff@bold\fontseries{b}\selectfont\fi%

```

```

212     #1%
213   }%
214 }

```

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

```

215 \newcommand\ff@rule
216   {\vrule height 0.6em depth 0.1em width 0em}

```

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

```

217 \iff@noframes\else
218   \RequirePackage{tcolorbox}
219   \newtcbox\ff@box{nobeforeafter,colframe=gray!80!white,
220     colback=gray!5!white,boxrule=0.01em,arc=0.1em,
221     boxsep=0.12em,left=0.05em,right=0.05em,top=0.02em,bottom=0.02em,
222     tcbox raise base}
223 \fi

```

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

```

224 \NewDocumentCommand\ff@x{v}{\ff{#1}}

```

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

```

225 \newcommand\ff[1]{%
226   \iff@noframes%
227   \ff@rule\ff@print{#1}%
228   \else%
229   \relax\ifmmode%
230     \ff@rule\ff@print{#1}%
231   \else%
232     \ff@box{\ff@rule\ff@print{#1}}%
233   \fi%
234   \fi%
235 }

```

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

```

236 \iff@novert\else
237   \catcode'\|active
238   \AtBeginDocument{\catcode'\|active\protected\def|{\ff@x|}}
239   \catcode'\| 12 %
240 \fi
241 \endinput

```

Change History

v0.1.0		
General: Initial version	4	
v0.10.0		
General: Package option <code>samepage</code> added, to prevent each <code>ffcode</code> block from breaking across pages.	4	
<code>\ffinput</code> : The <code>\ffinput</code> command added.	8	
v0.10.2		
General: We started to use <code>keepspaces</code> in order to guarantee correct visual rendering of multi-spaced content.	4	
v0.11.0		
<code>ffsave</code> : The <code>ffsave</code> environment added.	8	
v0.11.1		
General: Reserve <code>xleftmargin</code> for line numbers so they sit inside the column instead of protruding into the page margin.	4	
v0.12.0		
General: Static <code>\lstset</code> defaults moved from the per-environment hook to the preamble, so that user-supplied <code>\lstset</code> calls are no longer overwritten on every <code>ffcode</code> usage.	4	
<code>\ffinput</code> : Stopped <code>\ffinput</code> options from leaking into later listings.	8	
v0.12.1		
<code>ffsave</code> : The <code>ffsave</code> environment now creates the missing parent directory of its target file.	8	
v0.12.2		
General: Restore the original <code>\@vspace</code> inside listings, so <code>acmart</code> no longer warns about the vertical spacing that <code>listings</code> emits around every block.	4	
v0.13.0		
General: Package option <code>tmpfile</code> added, to configure the scratch file used to capture <code>ffcode</code> bodies.	4	
		<code>\ffcolumbreak</code> : Columns split by <code>\ffcolumbreak</code> are now sized by the longest line in each, instead of splitting <code>\linewidth</code> evenly.
		5
		Restore the original <code>\@vspace</code> inside every <code>\ffcolumbreak</code> column, so <code>acmart</code> no longer warns about the top-alignment strut in each column box.
		5
		The <code>\ffcolumbreak</code> marker added, to split an <code>ffcode</code> block into two or more side-by-side columns.
		5
	v0.15.0	
		<code>\ffcolumbreak</code> : Columns split by <code>\ffcolumbreak</code> now share the line equally by default, widening only a column that does not fit its equal share.
		5
	v0.15.1	
		<code>\ffcolumbreak</code> : Line numbers no longer restart in every <code>\ffcolumbreak</code> column when the <code>nocn</code> option is used.
		5
	v0.15.2	
		<code>\ffcolumbreak</code> : Only a line holding nothing but the escaped <code>\ffcolumbreak</code> marker now splits a block, so code that merely mentions it is no longer cut out.
		5
	v0.15.3	
		<code>\ffinput</code> : The <code>\ffinput</code> command now splits a file into columns at <code>\ffcolumbreak</code> markers, as the <code>ffcode</code> environment does.
		8
	v0.15.4	
		General: Package options are now true switches, so that setting <code>nonumbers</code> or any other option to <code>false</code> no longer turns the feature on.
		4
	v0.2.0	
		General: Package options <code>nonumbers</code> and <code>noframes</code> added.
		4
	v0.3.0	
		General: Package option <code>nocn</code> added.
		4

v0.4.0		added, to make all <code>\ff</code> pieces	
General: Package option <code>nobars</code>		look bolder than usual.	4
added.	4	Package option <code>sf</code> added, to	
v0.5.1		print all <code>\ff</code> pieces with	
<code>\ff@print</code> : Now, the command <code>ff</code>		<code>\sffamily</code>	4
ignores italic and bold and		v0.8.0	
always prints <code>\texttt</code> as it		<code>\ff</code> : The <code>\ff</code> command is now a	
should be.	8	normal command, not	
v0.6.0		verbatim.	9
General: Package option <code>novert</code>		v0.9.0	
added, to disable the		General: The <code>minted</code> package is	
redefinition of the vertical bar. . .	4	replaced by the <code>listings</code>	
We use <code>pgfopts</code> instead of		package.	4
<code>xkeyval</code>	4	v0.9.2	
v0.7.0		General: All lengths and sizes are	
General: Package option <code>bold</code>		in “em” instead of “pt”.	4

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		
<code>\@empty</code>	51, 203	
<code>\@ifnextchar</code>	174	
<code>\@ne</code> ..	77, 84, 87, 89, 103, 119, 149, 157	
<code>\@totalleftmargin</code> ..	162	
<code>\@vspace</code>	30, 144	
<code>\@vspace@orig</code>	28, 30, 143, 144	
<code>\@vspacer</code>	31, 145	
<code>\@vspacer@orig</code> ..	31, 145	
<code>_</code>	78	
<code>\^</code>	173, 179	
<code>\ </code>	237, 238, 239	
A		
<code>\active</code> ...	173, 237, 238	
<code>\addvspace</code>	167, 169	
<code>\advance</code>	77, 84, 87, 110, 118, 119, 157, 158	
<code>\AtBeginDocument</code> ..	238	
B		
<code>\begin</code>	182, 206	
<code>\bold</code>	2	
C		
<code>\c</code>	195	
<code>\c@lstnumber</code> ..	161, 165	
<code>\catcode</code> 77, 78, 173, 179, 237, 238, 239		
<code>\closein</code>	100	
<code>\color</code>	40, 44, 73	
<code>\count@</code>	76, 77	
<code>\cs</code>	190	
D		
<code>\def</code>	51, 73, 75, 79, 95, 105, 113, 116, 127, 135, 150, 162, 178, 180, 185, 238	
<code>\detokenize</code>	52	
<code>\dimexpr</code>	109, 125, 128, 138, 159	
E		
<code>\edef</code>	52, 85, 161	
<code>\end</code>	175, 207	
<code>\endinput</code>	241	
<code>\endlinechar</code>	78	
<code>\endminipage</code> ..	177, 186	
environments:		
<code>ffcode</code>	172	
<code>ffsave</code>	188	
<code>\expandafter</code> ..	85, 88, 102, 111, 153, 204	
<code>\ExplSyntaxOff</code>	200	
<code>\ExplSyntaxOn</code>	189	
F		
<code>\ff</code>	224, 225	
<code>\ff@allother</code>	75, 99	
<code>\ff@avail</code> ..	62, 125, 128, 137, 138, 159	
<code>\ff@begin</code>	174, 178	
<code>\ff@box</code>	217, 232	
<code>\ff@cbmarker</code>	52, 86	
<code>\ff@col</code> ...	112, 116, 135	
<code>\ff@cols</code> 111, 121, 160, 168		
<code>\ff@colsep</code>	73, 136	
<code>\ff@colwd</code> 59, 138, 140, 142		
<code>\ff@computelevel</code> 127, 166		
<code>\ff@denom</code> ..	68, 124, 125	
<code>\ff@file</code> 20, 98, 108, 148, 155, 182, 185		
<code>\ff@fixedcnt</code>	67, 114, 119, 123, 124, 132, 133	
<code>\ff@fixedsum</code>	66, 114, 118, 125	
<code>\ff@flat</code>	51, 85	
<code>\ff@flatline</code>	85, 86	
<code>\ff@grab</code> 71, 72, 162, 164		
<code>\ff@gutter</code> ..	64, 109, 162	
<code>\ff@level</code>	65, 117, 125, 128, 140	
<code>\ff@line</code>	80, 85	
<code>\ff@mbox</code>	70, 106	
<code>\ff@measure</code> ...	105, 163	
<code>\ff@mkdir</code>	204	
<code>\ff@natwd</code> 63, 109, 110, 112		
<code>\ff@nbreaks</code>	55, 87, 96, 101, 154, 157	
<code>\ff@ncols</code> ..	56, 123, 124, 128, 157, 159	
<code>\ff@next</code>	82, 92, 94	
<code>\ff@nlines</code>	54, 84, 89, 90, 96, 103	
<code>\ff@onepass</code> ...	113, 131	
<code>\ff@opts</code> ..	153, 180, 185	
<code>\ff@prev</code> 57, 136, 149, 168		
<code>\ff@prevcnt</code>	69, 129, 132, 133	
<code>\ff@print</code>	208, 227, 230, 232	
<code>\ff@read</code> 53, 80, 81, 98, 100		
<code>\ff@readloop</code> ..	79, 92, 99	
<code>\ff@render</code> ..	150, 176, 185	
<code>\ff@rule</code> 215, 227, 230, 232		
<code>\ff@savenum</code> ..	161, 165	
<code>\ff@scan</code>	95, 151	
<code>\ff@seg</code> ...	88, 102, 163	
<code>\ff@segs</code> 88, 97, 102, 163		
<code>\ff@sepwd</code> ..	60, 158, 159	
<code>\ff@start</code>	58, 89, 90, 97, 103	
<code>\ff@total</code>	61, 110, 137, 138, 160	
<code>\ff@x</code>	224, 238	
<code>ffcode</code> (env.)	172	
<code>\ffcolumbreak</code> ...	3, 49	
<code>\ffinput</code>	1, 183	
<code>ffsave</code> (env.)	188	
<code>\filename@area</code> 203, 204		
<code>\filename@parse</code> ...	202	
<code>\fontseries</code>	211	
G		
<code>\gdef</code>	97, 160	
<code>\global</code> 84, 87, 90, 96, 97, 110, 114, 118, 119, 160, 162, 165		
H		
<code>\hbox</code>	168	
<code>\hfuzz</code>	106	
<code>\hspace</code>	73, 74	
I		
<code>\ifdefined</code> 28, 31, 143, 145		
<code>\ifdim</code>	117, 137, 140	
<code>\ifeof</code>	81	
<code>\iff@bold</code>	2, 211	

<code>\iff@nobars</code>	7, 43	<code>\NewDocumentCommand</code>	224	S	
<code>\iff@nocn</code>	9, 46	<code>\newif</code> 2, 3, 4, 5, 6, 7, 8, 9		<code>\samepage</code>	2
<code>\iff@noframes</code> 5, 217, 226		<code>\newread</code>	53	<code>\selectfont</code>	211
<code>\iff@nonumbers</code>	6, 38	<code>\newsavebox</code>	70	<code>\setbox</code>	106
<code>\iff@novert</code>	8, 236	<code>\newtcbox</code>	219	<code>\setlength</code>	27
<code>\iff@samepage</code>	4, 177, 181, 184, 186	<code>\nobars</code>	2	<code>\sf</code>	2
<code>\iff@sf</code>	3, 210	<code>\nocn</code>	2	<code>\sffamily</code>	40, 210
<code>\ifmmode</code>	229	<code>\noexpand</code>	88, 102, 112	<code>\ShellEscape</code>	196, 198
<code>\ifnum</code>	77, 101, 123, 132, 136, 154	<code>\noframes</code>	2	<code>\str</code>	192, 195
<code>\ifx</code>	86, 203	<code>\noindent</code>	168, 181, 184	<code>\string</code>	52
J		<code>\nonumbers</code>	2	<code>\sys</code>	193
<code>\jobname</code>	21	<code>\novert</code>	2	T	
L		<code>\number</code>	138	<code>\textnormal</code>	209
<code>\l</code>	192, 195, 196, 198	<code>\numexpr</code>	89, 103, 124	<code>\the</code>	89, 103, 112, 161
<code>\lineskip</code>	27	O		<code>\tiny</code>	40
<code>\linewidth</code>	159, 181, 184	<code>\openin</code>	98	<code>\tmpfile</code>	2
<code>\loop</code>	77, 130	P		<code>\topsep</code>	167, 169
<code>\lst@AddToHook</code> 27, 29, 72		<code>\par</code>	167, 169	<code>\ttfamily</code>	35, 210
<code>\lst@maxwidth</code>	109	<code>\parbox</code>	142	U	
<code>\lstinputlisting</code>	107, 148, 155	<code>\pgfkeys</code>	10	<code>\unexpanded</code>	88, 102, 111
<code>\lstset</code>	33, 34, 35, 36, 37, 39, 44, 47, 136, 153	<code>\ProcessPgfpPackageOptions</code>	24	V	
M		<code>\protected</code>	238	<code>\vbox</code>	106
<code>\maxdimen</code>	106	R		<code>\VerbatimEnvironment</code>	182, 206
<code>\minipage</code>	181, 184	<code>\read</code>	80	<code>\vrule</code>	73, 216
N		<code>\relax</code>	71, 82, 89, 98, 103, 109, 118, 124, 125, 128, 138, 140, 159, 164, 165, 179, 229	<code>\vspace</code>	147
<code>\newcount</code>	54, 55, 56, 57, 58, 67, 68, 69	<code>\repeat</code>	77, 134	X	
<code>\newdimen</code>	59, 60, 61, 62, 63, 64, 65, 66	<code>\RequirePackage</code>	1, 25, 26, 49, 188, 218	<code>\xdef</code>	88, 102, 111
				Z	
				<code>\z@</code>	76, 96, 97, 101, 114, 129, 136, 154, 160, 168
				<code>\zap@space</code>	51