```
%%
%%   This is file `mcode.sty'
%%
%%   It is supposed to help you easily include MATLAB source code
%%   into LaTeX document,  but have it nicely highlighted, using
%%   the great listings package.
%%
%%   PLEASE NOTE that this package does nothing but save you from
%%   figuring out some configurations  in setting up the LISTINGS
%%   package. ALL the work is done by that package!  Thus, please
%%   refer your questions to the listings package documentation.
%%
%%   Usage: You have three ways of including your MATLAB code. As
%%   environment,  as inline object and directly from an external
%%   file.
%%
%%   1) Environment:
%%
%%          \begin{lstlisting}
%%             YOUR CODE HERE
%%          \end{lstlisting}
%%
%%
%%   2) Inline object:
%%
%%          Bla bla \mcode{CODEFRAGMENT} bla bla.
%%
%%
%%   3) Include external file (in environment form)
%%
%%          \lstinputlisting{YOUR-FILE.m}
%%
%%
%%   For your convenience this package has the following options:
%%
%%   - bw  if you intend to print the document (highlighting done
%%         via text formatting (bold, italic) and shades of gray)
%%
%%   - numbered  if you want line numbers
%%
%%   - autolinebreaks  if you want  the package to  automatically
%%          wrap your  code.  This is buggy as it  may well break
%%          break syntax and it  doesn't work well with comments.
%%          You REALLY should wrap your code manually.
%%
%%   - useliterate   if you want  some characters / relations  in
%%          your code to be replace with something more readable.
%%          Example: ~= becomes $\neq$, >= becomes $\geq$,  delta
%%          becomes $\delta$ and so on.
%%
%%   - framed  if you want a frame  around the source code blocks
%%
%%   - final  if you have  ``gloablly'' set the draft option, the
%%        listings package will  not output the code at all.  to
%%        force it to  do so anyway,  load this package with the
%%        final option (passes the ``final'' on to listings).
%%
%%   For example, you may use \usepackage[numbered,framed]{mcode}
%%   in your document preamble.
%%
```

```
%%   Note:  Inside code blocks you  can escape to LaTeX text mode
%%   using ¬ß...¬ß.  For ex. ¬ßtext and some math: $x^2$¬ß,  which is
%%   especially  useful  in comments  for putting  nicely typeset
%%   equations etc.  To get the same  colour/style as in the rest
%%   of the comment use \mcommentfont, i.e. ¬ß\mcommentfont $x^2$¬ß
%%
%%   To change the font used,  edit the first line in the "custo-
%%   mise below" section.  And feel free to  edit other things as
%%   well.  Refer to the documentation of the listings package to
%%   see what  else you could do.  If an extra small font  is re-
%%   quired,  use  {\fontfamily{pcr}\fontsize{3}{4.6}\selectfont}
%%   in the definition of \lstbasicfont.
%%
%%   Author:
%%      Florian Knorn | florian@knorn.org | www.florian-knorn.com
%%
%%   Version history:
%%      2.3  --  More keywords (thanks Dominik Wild!)
%%      2.2  --  Bugfix (thanks Willi Gerbig!)
%%      2.1  --  Finally automatic detection between end and end
%%      2.0  --  New options for line breaking and literate prog.
%%      1.8  --  Fixed typo in documentation regarding ¬ß...¬ß
%%      1.7  --  Added MATLAB block comment syntax %{ ..... %}
%%      1.6  --  Added some infos, dealing with keyword ``end''
%%      1.5  --  Tweaked check to see wether textcomp is loaded
%%      1.4  --  Fixed misconfig (mathescape now set to false)
%%      1.3  --  Purely cosmetic (tabs replaced by spaces)
%%      1.2  --  Added \lstset{showstringspaces=false}
%%      1.1  --  Added \mcode command and [final] option
%%      1.0  --  Release


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               D O N ' T    T O U C H    T H I S                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\def\fileversion{2.3}
\def\filedate{2012/08/31}

\typeout{-- Package: `mcode' \fileversion\space <\filedate> --}
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{mcode}[\filedate\space\fileversion]

% for bw-option
\newif\ifbw
\DeclareOption{bw}{\bwtrue}

% numbered option
\newif\ifnumbered
\DeclareOption{numbered}{\numberedtrue}

% final option
\newif\iffinal
\DeclareOption{final}{\finaltrue}

% autolinebreaks option
\newif\ifautolinebreaks
\DeclareOption{autolinebreaks}{\autolinebreakstrue}

% literate programming (replace certain characters/relations
\newif\ifuseliterate
```

```
\DeclareOption{useliterate}{\useliteratetrue}

% framed option
\newif\iframed
\DeclareOption{framed}{\framedtrue}

\DeclareOption*{% default
  \PackageWarning{mcode}{Unknown option `\CurrentOption' !}%
}
\ProcessOptions

\ifbw\typeout{ - settings optimized for printing (bw formating)}
\else\typeout{ - settings optimized for display (colour formating)}\fi
\ifnumbered\typeout{ - line numbering enabled}\else\fi
\ifuseliterate\typeout{ - literate programming (character replacements)
enabled}\else\fi
\ifautolinebreaks\typeout{ - automatic line breaking enabled (careful,
buggy!)}\else\fi
\iframed\typeout{ - framed listings}\else\fi

% This command allows you to typeset syntax highlighted Matlab
% code ``inline''.  The font size \small seems to look best...
\newcommand{\mcode}[1]{\lstinline[basicstyle=\lstbasicfont\small]|#1|}

% check if color command exists
\ifx\color\undefined%
  \RequirePackage{xcolor}%
\fi

% check if listings has been loaded
\ifx\lstset\undefined%
  \iffinal
    \RequirePackage[final]{listings}
  \else
    \RequirePackage{listings}
  \fi
\fi

% Check if textcomp has been loaded (this package is needed for
% upright quotes '' (instead of typographic ones `¬¥)...
\ifx\textquotesingle\undefined%
  \RequirePackage{textcomp}%
\fi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                C U S T O M I S E   B E L O W                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ------------------------------------------------------------------------
---
% default font
\def\lstbasicfont{\fontfamily{pcr}\selectfont\footnotesize}

% ------------------------------------------------------------------------
---
% matlat languate definition
\lstdefinelanguage{matlabfloz}{%
  alsoletter={...},%
  morekeywords={%                              % keywords
  break,case,catch,continue,elseif,else,end,%
```

```
    for,function,global,if,otherwise,persistent,%
      return,switch,try,while,methods,properties,%
        events,classdef,...},%
  comment=[l]\%,                                % comments
  morecomment=[l]...,                           % comments
  morecomment=[s]{\%\{}{\%\}},                  % block comments
  morestring=[m]'                               % strings
}[keywords,comments,strings]%

% ------------------------------------------------------------------------
---
% general definitions
\lstset{%
  basicstyle={\lstbasicfont},                   % set font
  showstringspaces=false,                       % do not emphasize spaces in
strings
  tabsize=4,                                    % number of spaces of a TAB
  mathescape=false,escapechar=¬ß,                 % escape to latex with ¬ß...¬ß
  upquote=true,                                 % upright quotes
  aboveskip={1.5\baselineskip},                 % a bit of space above listings
  columns=fixed                                 % nice spacing
}

% ------------------------------------------------------------------------
---
% define colours and styles
\ifbw % use font formating and gray 'colors'
      \def\mcommentfont{\color[gray]{.75}\itshape} %comments light gray and
italic
  \lstset{language=matlabfloz,                  % use our version of highlighting
    keywordstyle=\bfseries,                     % keywords in bold
    commentstyle=\mcommentfont,                 % comments
    stringstyle=\color[gray]{0.5}               % strings darker gray
  }
\else% notbw => use colors : )
      \def\mcommentfont{\color[rgb]{.133,.545,.133}} %comments in green
  \lstset{language=matlabfloz,                  % use our version of highlighting
    keywordstyle=\color[rgb]{0,0,1},            % keywords in blue
    commentstyle=\mcommentfont,                 % comments
    stringstyle=\color[rgb]{.627,.126,.941}     % strings in purple
  }
\fi%bw

% ------------------------------------------------------------------------
---
% automatic line breaking --- warning, this is buggy and
% doesn't break comments correctly!
\ifautolinebreaks
      \newsavebox{\lbreakdots}\sbox{\lbreakdots}{\lstbasicfont\mcommentfont...}
      \lstset{breaklines=true,breakatwhitespace=true,prebreak=\usebox{\lbreakdot
s}}
\fi

% ------------------------------------------------------------------------
---
% literate replacements
% the following is for replacing some matlab relations like >= or ~=
% by the corresponding LaTeX symbols, which are much easier to read ...
\ifuseliterate
      \lstset{%
```

```
            literate=%
                  {~}{{$\neg$}}1 %               \neg
                  {<=}{{\tiny$\leq$}}1 %         \leq
                  {>=}{{\tiny$\geq$}}1 %         \geq
                  {~=}{{\tiny$\neq$}}1 %         \neq
                  {delta}{{\tiny$\Delta$}}1 %    \Delta
                  {(end)}{\lstbasicfont (end)}{5} % black ``end'' when indexing
last vector element
                  {({ }end)}{\lstbasicfont ({ }end)}{6}
                  {(end{ })}{\lstbasicfont (end{ })}{6}
                  {({ }end{ })}{\lstbasicfont ({ }end{ })}{7}
                  {:end}{\lstbasicfont :end}{4}
                  {:{ }end}{\lstbasicfont :{ }end}{5}
                  {end:}{\lstbasicfont end:}{4}
                  {end{ }:}{\lstbasicfont end{ }:}{5}
                  {,end}{\lstbasicfont ,end}{4}
                  {,{ }end}{\lstbasicfont ,{ }end}{5}
      }
\else
      \lstset{%
            literate=%
                  {(end)}{\lstbasicfont (end)}{5} % black ``end'' when indexing
last vector element
                  {({ }end)}{\lstbasicfont ({ }end)}{6}
                  {(end{ })}{\lstbasicfont (end{ })}{6}
                  {({ }end{ })}{\lstbasicfont ({ }end{ })}{7}
                  {:end}{\lstbasicfont :end}{4}
                  {:{ }end}{\lstbasicfont :{ }end}{5}
                  {end:}{\lstbasicfont end:}{4}
                  {end{ }:}{\lstbasicfont end{ }:}{5}
                  {,end}{\lstbasicfont ,end}{4}
                  {,{ }end}{\lstbasicfont ,{ }end}{5}
      }
\fi%literates

% ------------------------------------------------------------------------
---
% line numbering
\ifnumbered% numbered option
  \lstset{%
    numbersep=3mm, numbers=left, numberstyle=\tiny, % number style
  }
\fi

\iframed%   framed option
  \lstset{%
    frame=single,                                 % frame
  }
  \ifnumbered%
    \lstset{%
      framexleftmargin=6mm, xleftmargin=6mm   % tweak margins
    }
  \fi
\fi

\endinput
%% End of file `mcode.sty'.
```