

```

%%
%% 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.
%%
%% Usage: Include your MATLAB source code by using
%%
%% \begin{lstlisting}
%% YOUR CODE HERE
%% \end{lstlisting}
%%
%% or as an inline object via \mcode{YOURCODE}.
%%
%% 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
%%
%% - framed if you want a frame around the source code blocks
%%
%% - final if you have ``gloabllly" 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).
%%
%% Example of use: \usepackage[numbered,framed]{mcode}
%% in your document preamble.
%%
%% Note: inside code blocks you can 'escape' to LaTeX math mode
%% by using $ YOUR LATEX CODE $, which is especially useful in
%% comments...
%%
%% Another feature of the listings package is that you can re-
%% place certain strings by LaTeX strings; this is used for
%% some relation symbols, see below...
%%
%% Mat Odijk pointed this out, you may include entire m-files
%% using the command \lstinputlisting{YOUR-FILE.m}. Thanks for
%% the tip!
%%
%% Feel free to edit things, and refer to the listings package
%% documentation for more infos.
%%
%% If you have any questions, feel free to ask: floz@gmx.de
%%
%% Unsolved problem: long lines of code that are wrapped with
%% '...', and things thereafter being comments.....
%% but i'm working on it ;-)
```

```
%% Author: Florian Knorn, floz@gmx.de
%%
%% Version history:
%% 1.2 -- Added \lstset{showstringspaces=false}
%% 1.1 -- Added \mcode command and [final] option
%% 1.0 -- Release
```

```
\def\fileversion{1.2}
\def\filedate{2005/11/17}
```

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

```
% for bw-option
\newif\ifbw
\DeclareOption{bw}{\bwtrue}
\ifbw\typeout{mcode: settings optimized for printing!}
\else\typeout{mcode: settings optimized for display!}\fi
```

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

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

```
% for framed option
\newif\ifframed
\DeclareOption{framed}{\framedtrue}
```

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

```
% with this command, you can typeset syntax highlighted mcode ``inline",
% for example when you talk about \mcode{for}--loops ...
\newcommand{\mcode}[1]{\lstinline[basicstyle=\lstbasicfont]#1|}
```

```
% check if color command exists
\ifx\color\undefined%
  \RequirePackage{color}%
\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\textasciigrave\undefined%
  \RequirePackage{textcomp}%
\fi

%%%%%%%%%%
% CONFIGS --- CUSTOMIZE HERE %
%%%%%%%%%%

% define the wanted font for all highlightings here
\def\lstbasicfont{\fontfamily{pcr}\selectfont}

% now let's define our own version of matlab highlighting
\lstdefinelanguage{matlabflox}{%
  alsoletter={...},%
  morekeywords={% % keywords
    break,case,catch,continue,elseif,else,end,for,function,global,%
    if,otherwise,persistent,return,switch,try,while,....,
    classdef,properties,methods},%
  comment=[l]\%,% % comments
  morecomment=[l]...,% % comments
  morestring=[m]',% % strings
}[keywords,comments,strings]%

\ifbw % use font formating and gray 'colors'
  \lstset{language=matlabflox, % use our version of highlighting
    keywordstyle=\bfseries, % keywords in bold
    commentstyle=\color[gray]{0.6}\itshape, % comments light gray and italic
    stringstyle=\color[gray]{0.5} % strings darker gray
  }
\else% notbw => use colors : )
  \lstset{language=matlabflox, % use our version of highlighting
    keywordstyle=\color[rgb]{0,0,1}, % keywords
    commentstyle=\color[rgb]{0.133,0.545,0.133}, % comments
    stringstyle=\color[rgb]{0.627,0.126,0.941} % strings
  }
\fi%bw

\lstset{%
  basicstyle={\lstbasicfont\footnotesize}, % use font and smaller size
  showstringspaces=false, % do not emphasize spaces in strings
  tabsize=4, % number of spaces of a TAB
  mathescape=true,escapechar=§, % escape to latex with §...§
  upquote=true, % upright quotes
  aboveskip={1.5\baselineskip}, % a bit of space above
  columns=fixed, % nice spacing
  %
  % the following is for replacing some matlab relations like >= or ~=
  % by the corresponding LaTeX symbols, which are much easier to read ...

```

```

%% literate=%
%%      {\~}{\{\$neg$}}1 %          \neg
%%      {\<=}{\{\tiny$\leq$}}1 %    \leq
%%      {\>=}{\{\tiny$\geq$}}1 %    \geq
%%      {\~={}{\{\tiny$\neq$}}1 %    \neq
%%      {\delta}{\{\tiny$\Delta$}}1% \Delta
}

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

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

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

```