

# tcvprgm — Simple commands for typesetting programming things.\*

Taylor Venable<sup>†</sup>

Released ?

1 `\package`

We require the `pdfpages` package to use the `minipage` environment properly within a PDF document.

2 `\RequirePackage{pdfpages}`

`\empty` A macro which can be expanded for use with `\if` and which is empty. Useful for checking if things are undefined.

3 `\def\empty{}`

`\flag` Typeset a normal program or Tcl command flag; the first argument is the name of the option, the second optional argument is the metasyntactic variable to describe the parameter of that flag.

`\flag*` Same as the above, but is used to represent an optional flag by placing the result inside square brackets.

4 `\newcommand{\flag}{\@ifstar{\flag@optional}{\flag@normal}}`

5

6 `\newcommand{\flag@normal}[2][{}]{%`

7 `\def\tcv@value{#1}%`

8 `{\tt -#2}\if\tcv@value\empty\else\ {\it #1}\fi%`

9 `}`

10

11 `\newcommand{\flag@optional}[2][{}]{%`

12 `{\tt []\flag@normal[#1]{#2}{\tt}}%`

13 `}`

`\fn` Typeset a function name inline.

`\fn*` Typeset a specially offset function inside a block.

14 `\newcommand{\fn}{\@ifstar\fn@long\fn@short}`

15 `\newcommand{\fn@short}[1]{\textit{#1}\texttt{()}}`

---

\*This file describes version ?, last revised ?.

<sup>†</sup>E-mail: [taylor@metasyntax.net](mailto:taylor@metasyntax.net)

```

16 \newcommand{\fn@long}[2]{%
17   \noindent\colorbox{grey}{\begin{minipage}{\textwidth-6pt}%
18     {\sffamily \iffnbold\bfseries\fi #1}{\tt{}#2{\tt}}}%
19   \end{minipage}}\vskip1ex\hrule\vskip1ex\noindent\relax%
20 }

21 \newif\iffnbold
22 \fnboldtrue

```

`\var` Typeset a variable name, or function parameter.

`\var*` Typeset an optional variable or argument. The same as the above, but it is placed within square brackets.

```

23 \newcommand{\var}{\@ifnextchar*{\var@optional}{\var@normal}}
24 \def\var@optional*{\var@optionalx}
25 \newcommand{\var@normal}[2][{}]{%
26   \def\tcv@type{#1}%
27   \def\array##1{\tffamily[]##1{\tffamily}}}%
28   \textit{#2}\ifx\tcv@type\empty\else{} : \tcv@type\fi%
29 }}
30 \newcommand{\var@optionalx}[2][{}]{%
31   \def\tcv@type{#1}%
32   \def\array##1{\tffamily[]##1{\tffamily}}}%
33   {\tffamily[]\textit{#2}\ifx\tcv@type\empty\else{} : \tcv@type\fi{\tffamily}}%
34 }}

```

`\anglize` Typeset something in italics inside angle brackets. The result looks like this: *(foo)*.

```

35 \newcommand{\anglize}[1]{\ensuremath{\langle}\textit{#1}\ensuremath{\rangle}}
36 \newenvironment{testcase}[1]{%
37   \begin{Sbox}%
38     % Subtract 1pt in width to avoid overfull hbox warnings.
39     \begin{minipage}{\textwidth-1pt}%
40       \setlength{\fboxsep}{4pt}%
41       % 4pt of padding on either side = 8pt difference
42       \noindent\colorbox{grey}{\parbox{\textwidth-8pt}{\textsf{\textbf{#1}}}}%
43       \list{}{\leftmargin4pt\rightmargin\leftmargin\topsep4pt}\item\relax
44 }{%
45   \vspace{4pt plus 0pt minus 0pt}
46   \endlist
47   \end{minipage}%
48   \end{Sbox}%
49   \setlength{\fboxsep}{0pt}%
50   \noindent\fbbox{\parbox{\textwidth}{\TheSbox}}%
51 }

52 \newcommand{\cmd}{\@ifnextchar*{\cmd@nohrule}{\cmd@hrule}}
53 \newcommand{\cmd@hrule}[4][{}]{%
54   \def\tcv@namespace{#1}%
55   \vskip2ex\noindent\colorbox{grey}{\begin{minipage}{\textwidth-6pt}%
56     \begin{group}

```

```

57 \if\tcv@namespace\empty\else\tcv@namespace::\fi%
58 {\bfseries #2}%
59 \endgroup #3 \textit{#4}%
60 \end{minipage}}\vskip1ex\hrule\vskip1ex\noindent\relax%
61 }
62 \def\cmd@nohrule*#1#2#3#4{%
63 \def\tcv@namespace{#1}%
64 \vskip2ex\noindent\colorbox{grey}{\begin{minipage}{\textwidth-6pt}%
65 \begingroup
66 \if\tcv@namespace\empty\else\tcv@namespace::\fi%
67 {\bfseries #2}%
68 \endgroup\ \texttt{#3} #4%
69 \end{minipage}}\noindent\relax%
70 }

```

`\scmfn` Print a Scheme function name inline. The argument is the function name. For both this command and `\scmfn*` below, you may use `\rightarrow` to typeset a right arrow, as is commonly used in Scheme function names.

`\scmfn*` Typeset a Scheme function declaration as a special box in the page. This calls it out for reference, like `\fn*` does. The first argument is the function name, the second is whatever should be printed after that (i.e. arguments, keywords, etc).

```

71 \newcommand{\scmfn}{%
72 \def\rightarrow{\fontspec{Latin Modern Mono Prop}\rightarrow}%
73 \@ifstar\scmfn@long\scmfn@short%
74 }
75 \newcommand{\scmfn@short}[1]{\texttt{()}\textit{#1}\texttt{()}}
76 \newcommand{\scmfn@long}[2]{%
77 \noindent\colorbox{grey}{\begin{minipage}{\textwidth-6pt}%
78 {\texttt{()}\{\sffamily #1} #2\texttt{()}}}%
79 \end{minipage}}\vskip1ex\hrule\vskip1ex\noindent\relax%
80 }

```

`\scmvar` Print a variable in Scheme. The difference between this and regular `\var` is that the optional type argument is typeset with angle brackets around it.

```

81 \newcommand{\scmvar}{\@ifstar{\scmvar@optional}{\scmvar@normal}}
82 \newcommand{\scmvar@normal}[2][{}]{%
83 \def\tcv@type{#1}%
84 \def\array##1{\ttfamily []##1{\ttfamily}}}%
85 \textit{#2}\ifx\tcv@type\empty\else{} : \texttt{<}\textsf{\tcv@type}\texttt{>}\fi%
86 }}
87 \newcommand{\scmvar@optional}[2][{}]{%
88 \def\tcv@type{#1}%
89 \def\array##1{\ttfamily []##1{\ttfamily}}}%
90 {\ttfamily []\textit{#2}\ifx\tcv@type\empty\else{} : \texttt{<}\textsf{\tcv@type}\texttt{>}\fi%
91 }}
92 \AtBeginDocument{
93 \def\~{\discretionary}{-}{-}}
94 \def\.\{\discretionary}{.}{.}}
95 }

```

```
96 \def\IndexEntry#1#2{\phantomsection\label{fn:#1:#2}\index{#1@\texttt{#1}!#2@\texttt{#2}}}  
97 \newenvironment{params}[1]{%  
98 \begin{description}[\topsep0pt\compact\setleftmargin{#1}\setlabelstyle{\mdseries}]%  
99 }{%  
100 \end{description}%  
101 \vskip1ex\hrule\vskip1ex%  
102 }  
103 % \begin{macrocode}  
104 \end{macrocode}
```