

The `skeycommand` Package[☆]

Version 0.2

Ahmed Musa ✉
Preston, Lancashire, UK

20th May 2010

SUMMARY

The `skeycommand` package provides tools for defining L^AT_EX-style commands and environments using parameters and keys together. The advantages of keys over parameters include the facts that the former aren't limited to nine but can rise as desired by the user, and keys are much easier to match to their values than parameters to arguments, especially if the parameters are many. Moreover, keys can have natural functions. The design approach and user interfaces in the `skeycommand` package differ from those found in the `keycommand` package. This package also provides the `\newtwoptcmd` and `\newtwoptenv` macros for defining new commands and environments with two options/optional arguments. At both key command definition and invocation times there is no reference by the user to the semantics of key parsing and management. All the complex semantics and calculations involved in defining and setting keys are transparent to the user. The user of the `skeycommand` package has access to some (but not all) of the machinery of `xkeyval` and `skeyval` packages (including the pointer mechanism) at the much lesser cost of worrying only about the key names and their values. Native boolean keys are automatically recognized and handled appropriately. However, because of the need to keep the user interface simple, toggle boolean keys aren't available in this package; neither are choice keys.

LICENSE

This work (i.e., all the files in the `skeycommand` bundle) may be distributed and/or modified under the conditions of the L^AT_EX Project Public License (LPPL), either version 1.3 of this license or any later version.

The L^AT_EX Project Public License maintenance status of this software is “author-maintained”. This software is provided “as it is,” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

© MMX

[☆]The package is available at <http://www.ctan.org/tex-archive/macros/latex/contrib/skeycommand/>.

CONTENTS

1 Package options	2	2.3 Commands with two optional arguments	3
2 User interfaces	2	2.3.1 Invoking commands and environments	4
2.1 Defining new key commands and environments	2		
2.2 Invoking new key commands and environments	3	3 Examples	4

1 PACKAGE OPTIONS

The package has only one option (namely, `verbose`) and can be invoked at the time of loading the package or via the `\skeycommand` macro. The option `verbose` is a boolean, initially set to false (i.e., its complement, `silent`, is true by default). Setting `silent` to `false` is tantamount to setting `verbose` to `true`.

Example

```

1 % In style files:
2 \RequirePackage[verbose=true or false or on or off]{skeycommand}

3 % In document files:
4 \usepackage[verbose=true or false or on or off]{skeycommand}

5 % In all cases:
6 \skeycommand{verbose=true or false or on or off}

```

If you enter the boolean `verbose` (or `silent`) without value, the value is assumed to be `true`. The `verbose` option is simply passed on to the `skeyval` package to log informational messages in the transcript file. The major task of key parsing for the `skeycommand` package is undertaken by the `skeyval` package.

2 USER INTERFACES

2.1 Defining new key commands and environments

The user interfaces for defining new key commands and environments are as follows:

Macro

```

7 \newkeycmd<cmd><mprefix><(keyval)>[<narg>][<default1>]{<defn>}
8 \newkeycmd*<cmd><mprefix><(keyval)>[<narg>][<default1>]{<defn>}

9 \renewkeycmd<cmd><mprefix><(keyval)>[<narg>][<default1>]{<defn>}
10 \renewkeycmd*<cmd><mprefix><(keyval)>[<narg>][<default1>]{<defn>}

11 \newkeyenv<env><mprefix><(keyval)>[<narg>][<default1>]{<defn>}
12 \newkeyenv*<env><mprefix><(keyval)>[<narg>][<default1>]{<defn>}

13 \renewkeyenv<env><mprefix><(keyval)>[<narg>][<default1>]{<defn>}
14 \renewkeyenv*<env><mprefix><(keyval)>[<narg>][<default1>]{<defn>}

```

Here, `<cmd>` is the new control sequence; `<env>` is the new environment name; `<mprefix>` is the prefix for macros deriving from the defined keys whose values will be used in the new

command or environment (this is called the *macro prefix* in the parlance of keys); $\langle\text{keyval}\rangle$ is the key-value list [e.g., ($\text{keya}=\text{valuea}$, $\text{keyb}=\text{valueb}$)]; $\langle\text{narg}\rangle$ is the number of arguments for the new command or environment (excluding the keys), as you would enter it in $\backslash\text{newcommand}$ and $\backslash\text{newenvironment}$; $\langle\text{default1}\rangle$ is the default value for your optional argument (normally the first argument in $\backslash\text{newcommand}$ and $\backslash\text{newenvironment}$); $\langle\text{defn}\rangle$ is the replacement text (as in $\backslash\text{newcommand}$ and $\backslash\text{newenvironment}$). If you don't supply the optional $\langle\text{mprefix}\rangle$, the package will use the first three letters of the key command or environment name, excluding the escape character but including an added “at sign” (@). The aim of default “at sign” is to aid the visual separation of key names from macro prefixes.

Please note the angle bracket surrounding $\langle\text{mprefix}\rangle$, and the parenthesis surrounding $\langle\text{keyval}\rangle$ in the above syntaxes. The $\langle\text{mprefix}\rangle$ can't be empty (i.e., don't enter $\langle\rangle$) because it will be used by the package to build unique names for the macros that will hold the key values. You can choose not to enter anything for $\langle\text{mprefix}\rangle$, and the package will happily use the default prefix $\langle\text{xxx@}\rangle$, where “xxx” represents the first three letters of the new command or environment name. Also, $\langle\text{keyval}\rangle$ can't be empty: if it was empty, then we should ask: why use key commands instead of L^AT_EX's $\backslash\text{newcommand}$ and $\backslash\text{newenvironment}$?

In $\langle\text{defn}\rangle$, you refer to your arguments in the normal way. You refer to the values of the keys using macros whose first three characters (after the escape character) are the $\langle\text{mprefix}\rangle$ or, if $\langle\text{mprefix}\rangle$ is not supplied, the first three letters of the declared key command (excluding the escape character). The family name of the keys defined via a key command is the key command name itself (without the escape character). The package uses this internally in developing the keys. The key prefix is always “KV”. If any of your key values contains parenthesis, simply enclose it in braces, to avoid confusing it with $\langle\text{keyval}\rangle$ list.

The starred variants (\star) give “short” macros, while the unstarred variants yield “long” macros.

Note: The optional $\langle\text{mprefix}\rangle$ will be useful if you fear clashes with previously defined key commands. Although, to be defined, key commands must be definable, two key commands may have their first three or four characters identical, thereby leading to clashes of their key-value prefixes.

2.2 Invoking new key commands and environments

The syntaxes for calling new key commands and environments are as follows:

	Macro
15	$\backslash\text{cmd}[\langle\text{arg1}\rangle]\{\langle\text{arg2}\rangle\}\dots\{\langle\text{argn}\rangle\}(\langle\text{keyval}\rangle)$
16	$\backslash\text{begin}\{\text{env}\}[\langle\text{arg1}\rangle]\{\langle\text{arg2}\rangle\}\dots\{\langle\text{argn}\rangle\}(\langle\text{keyval}\rangle)$
17	environment body
18	$\backslash\text{end}\{\text{env}\}$

where $\backslash\text{cmd}$ and env have been previously defined using key command and key environment. You refer to your arguments using parameter number one $\#1$ onwards, up to a maximum of $\#8$ (yes, $\#8$, not $\#9$). Here, $\langle\text{keyval}\rangle$ (including the parenthesis) are optional arguments: you can omit them if you want to use the values of the keys set at key command definition time. Using keys is preferable to using parameters: you don't have to match parameters to arguments and, in principle, there is no limit to the number of keys that are permissible.

2.3 Commands and environments with two optional arguments

The `skeycommand` package uses the following macros internally. They can be used to define new commands and environments with two optional arguments. Their philosophy, intent,

and use syntaxes differ from those of the `twoopt` package. They may be useful to some users, although I recommend the use of the above key commands.

Macro

```

19 \newtwooptcmd⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}
20 \newtwooptcmd*⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}

21 \renewtwooptcmd⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}
22 \renewtwooptcmd*⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}

23 \newtwooptenvirion⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}
24 \newtwooptenvirion*⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}

25 \renewtwooptenvirion⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}
26 \renewtwooptenvirion*⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}

```

⟨narg⟩ is the total number of arguments, including the first and second optional arguments. Where are the second optional arguments here, you might be wondering? The second optional argument is usually empty and doesn't appear at command definition time. The second optional argument isn't the second argument of your command (as in `twoopt` package), but the last. At command invocation, if you don't supply a value for the second optional argument, the command will assume it to be empty. But how do you supply a value for the second optional argument? The next section shows how.

2.3.1 Invoking commands and environments with two optional arguments

The syntaxes for calling commands and environments with two optional arguments are as follows:

Macro

```

27 \cmd[⟨1st optarg⟩]{⟨arg2⟩}...{⟨argn⟩}(⟨2nd optarg⟩)

28 \begin{env}[⟨1st optarg⟩]{⟨arg2⟩}...{⟨argn⟩}(⟨2nd optarg⟩)
29   environment body
30 \end{env}

```

If ⟨2nd optarg⟩ is empty at command or environment invocation, the command or environment will assume it to be empty. Now you can see the conceptual link between `\newtwooptcmd` (and friends) and `\newkeycmd` (and friends).

3 EXAMPLES

Example

```

31 % The following is a macro of 3 parameters and 4 keys:

32 \newkeycmd*\demomacro(name=Steve,height=1.60m,weight=75kg,
33   tested=true)[3][Registered]{%
34   \def\x{#1}\def\y{#2}\def\z{#3}%
35   \ifdem@tested
36     \edef\cleared{\dem@name}%
37   \else
38     \let\cleared\relax
39   \fi
40 }
41 % \dem@name will hold the value supplied for 'name'

```

```

42 % by the user of \demomacro. 'dem' is from 'demomacro'.
43 % Notice the LaTeX-like syntax of this command.
44 % The user doesn't have to bother about the nitty-gritty
45 % of key infrastructure.

46 % You can use the following statement to instruct the user
47 % to always supply value for 'name' in \demomacro macro:

48 \uservaluekeys[KV]{demomacro}{name}

```

Instead of using the macro `\uservaluekeys`, you can use the full pointer system of the `skeyval` package. For example, the following requires the user to supply a value for 'name' and the package will save that value for some later use:

Example

```

49 \newkeycmd*\demomacro(\uservalue\savevalue{name}=Steve,
50   height=1.60m,wieght=75kg,tested=true)[3][Registered]{%
51   \def\x{#1}\def\y{#2}\def\z{#3}%
52   \edef\xxx{\dem@name}%
53 }

54 % User now calls the \demomacro macro:
55 \demomacro[data1]{data2}{data3}(name,height=1.55m,wieght=55kg,
56   tested=true)
57 → Error: no value supplied for 'name'

```

You can use the saved value of key `name` within `\demomacro`. Suppose, oddly, that the key `height` can use the value of the key `name`, then we can do

Example

```

58 \demomacro[data1]{data2}{data3}(name=John,
59   height=\usevalue{name},wieght=55kg,tested=true)

```

If in `\demomacro` you don't supply a key-value pair, the macro will use the above defaults:

Example

```

60 \demomacro[data1]{data2}{data3}(name=John,% height missing here.
61   weight=55kg,tested=true)

```

Example

```

62 \newkeycmd*\firstmacro<sk@>(name=Steve,
63   height=1.6m)[8][xxx]{%
64   \noindent Name: \sk@name\newline
65   Height: \sk@height\newline
66   Details: #1#2#3#4#5#6#7#8%
67   \newline\newline
68 }

69 \begin{document}
70 \firstmacro[1]{-2}{-3}{-4}{-5}{-6}{-7}{-8}%
71   (name=John {(Winner)},height=1.54m)
72 \end{document}

```

Example

```

73 \newkeyenviron*{vdescription}<skv@>(labelwidth=5pt,
74   labelsep=5pt)[2][\qqquad
75   {\begin{list}{}{\redef*\makelabel##1{\sffamily ##1:\hfil}}%
76     \settowidth\labelwidth{\makelabel{#1}}%
77     \SKV@dimdef\leftmargin{\labelwidth+\skv@labelwidth
78       +\labelsep+\skv@labelsep}}%
79     \item[Description Preamble] #2%
80   }{\end{list}\ignorespacesafterend}

81 \begin{document}
82 \begin{vdescription}[Description Postamble]%
83   {${\star\star\star}$}(labelwidth=10pt,labelsep=10pt)
84   \item[Item 1] xxx
85   \item[Item 2] yyy
86   \item[Description Postamble] ${\langle}$End of my
87     environment${\rangle}$
88 \end{vdescription}
89 \end{document}

```

Example

```

90 \def\@barbara{An author writing an article for publication
91   in TUGboat is encouraged to create it on a computer file
92   and submit it on magnetic tape.}
93 \def\barbara{BARBARA BEETON,\,\emph{How to Prepare a File For
94   Publication in TUGboat}} (1981)}
95 \def\@hieronymus{The printer should refuse to employ wandering
96   men, foreigners who, after having committed some grievous
97   error, can easily disappear and return to their own country.}
98 \def\hieronymus{HIERONYMUS HORNSCHUCH (1608)}
99 \def\@elgar{If I write

100 % The macros \@barbara, \barbara, etc. are just shorthands:
101 % you can enter their contents directly in key commands,
102 % as we shall see later.

103 \newkeyenviron{Quote}<mp@>(leftmargin=\leftmargin,
104   rightmargin=\rightmargin,mode=false,whoby=\null,
105   source=\null){\begin{list}{}{%
106     \setlength\leftmargin{\mp@leftmargin}%
107     \setlength\rightmargin{\mp@rightmargin}%
108   }}%
109   \item[]\makebox[0pt][r]{‘}%
110 }{%
111   \unskip\makebox[0pt][l]{’}%
112   \item[] \flushright\mp@whoby
113   \item[] \flushleft\small Source: \mp@source
114   \end{list}
115   \endgraf\vskip\baselineskip
116 }
117 \usepackage{lipsum}
118 \lipsum[1]
119 \begin{Quote}(leftmargin=30pt,rightmargin=30pt,mode=false,
120   whoby=\barbara,source={The \TeX Book})

```

```

121     {\ifmp@mode\color{red}\else\color{blue}\fi \@barbara}
122 \end{Quote}

123 \lipsum[1]
124 \begin{Quote}(leftmargin=20pt,rightmargin=20pt,mode=true,
125     whoby=\hieronymus,source={The \TeX Book})
126     {\ifmp@mode\color{red}\else\color{blue}\fi \@hieronymus}
127 \end{Quote}

128 \lipsum[1]
129 \begin{Quote}(leftmargin=40pt,rightmargin=40pt,mode=false,
130     whoby={Edward Elgar},source={Letter to A.\ J.\ Jaeger (1898)})
131     {\ifmp@mode\color{red}\else\color{blue}\fi
132     If I write a tune you all say it's commonplace---if I
133     don't, you all say it's rot.%
134     }%
135 \end{Quote}

136 \begin{Quote}(leftmargin=40pt,rightmargin=40pt,mode=false,
137     whoby={Albert Einstein},source={The World As I See It})
138     {\ifmp@mode\color{red}\else\color{blue}\fi
139     If you want to find out anything from the theoretical physicists
140     about the methods they use, I advise you to stick closely to
141     one principle: don't listen to their words, fix your attention
142     on their deeds.%
143     }%
144 \end{Quote}

```

Example

```

145 \usepackage{lipsum}
146 \newcounter{notecnt}
147 \def\noteparameters{\labelsep=\notelabelsep
148     \itemindent=\noteitemindent \leftmargin=\noteleftmargin
149     \rightmargin=\noterightmargin \labelwidth=\notelabelwidth}

150 \newkeyenvron*{note}<note>(labelsep=8pt,itemindent=8pt,
151     leftmargin=\parindent,rightmargin=\parindent,labelwidth=0pt,
152     preskip=0ex)[1][\baselineskip]%
153     {\begin{list}{\textsc{Note}~\arabic{notecnt}: }%
154         {\noteparameters\usecounter{notecnt}}}%
155         \vskip#1}%
156     {\end{list}\ignorespacesafterend}

157 \begin{document}
158 \noindent\lipsum[1]
159 \begin{note}[\notepreskip](labelsep=8pt,itemindent=8pt,
160     leftmargin=30pt,rightmargin=30pt,labelwidth=0pt,preskip=2ex)
161 \item \lipsum[1]
162 \item \lipsum[1]
163 \end{note}
164 \end{document}

```

Example

```

165 \def\sitation{}
166 \def\sitparameters{\leftmargin=\sit@leftmargin
167   \rightmargin=\sit@rightmargin}
168 \newbox\sitname
169 \renewkeyenvron*{sitation}(\leftmargin=\parindent,
170   rightmargin=\parindent,nolinebreak=1)[2][1]%
171   {\def\cnolinebreak{#1}\sitparameters\sbox\sitname
172     {\textit{#2}}}%
173     \begin{quote}%
174     }%
175     {\hspace*{\fill}\nolinebreak[\cnolinebreak]%
176       \quad\hspace*{\fill}\finalhyphendemerits\z@
177       \box\sitname
178       \end{quote}\ignorespacesafterend}

179 \begin{document}
180 \begin{sitation}[\sit@nolinebreak]%
181   {Hieronymus Hornschuch~(1608)}%
182   (\leftmargin=30pt,rightmargin=30pt,nolinebreak=2)
183   The printer should refuse to employ wandering men,
184   foreigners who, after having committed some grievous error,
185   can easily disappear and return to their own country.
186   \end{sitation}

187 \begin{sitation}{William Stanley Jevons~(1878)}%
188   Some compositors still object to work in offices where
189   type-composing machines are introduced. They are all afraid
190   that if the work is done too easily and rapidly, they will
191   not be wanted to do it. Justifying and page makeup can't be
192   done profitably by machines, so the employees needn't fear
193   losing their jobs.
194   \end{sitation}
195 \end{document}

```

Example

```

196 \newkeyenvron*{vdescription}(labelwidth=5pt,
197   labelsep=5pt)[2][\qqquad]%
198   {\begin{list}{}{\redef*\makelabel##1{\sffamily ##1:\hfil}}%
199     \settowidth\labelwidth{\makelabel{#1}}%
200     \SKV@dimdef\leftmargin{\labelwidth+\vde@labelwidth
201       +\labelsep+\vde@labelsep}}%
202     \item[Description Preamble] #2%
203   }\end{list}\ignorespacesafterend}

204 \begin{document}
205 \begin{vdescription}[Description Postamble]{+++xxx+++}%
206   (labelwidth=10pt,labelsep=5pt)
207   \item[Item 1] xxx
208   \item[Item 2] yyy
209   \item[Description Postamble] $\langle$End of my
210     environment$\rangle$
211 \end{vdescription}

```



```
212 \end{document}
```

Example

```
213 \renewtwoptenviro*{vdescription}[3][\qqquad]
214   {\begin{list}{}{\redef*\makelabel##1{\sffamily ##1:\hfil}}%
215     \settowidth\labelwidth{\makelabel{#1}}%
216     \SKV@dimdef\leftmargin{\labelwidth+\labelsep+#2}}%
217     \item[Description Preamble] #3%
218   }{\end{list}\ignorespacesafterend}

219 \begin{document}
220 \begin{vdescription}[Description Postamble]{4cm}({(Begin
221   environment no.\ 1)})
222   \item[Item 1] xxx
223   \item[Item 2] yyy
224   \item[Description Postamble] (End of environment no.\ 1)
225 \end{vdescription}
226 \end{document}
```

Example

```
227 \newtwoptenviro*{udescription}[3][\hspace{1cm}]
228   {\begin{list}{}{\redef*\makelabel##1{\sffamily ##1:\hfil}}%
229     \settowidth\labelwidth{\makelabel{#1}}%
230     \SKV@dimdef\leftmargin{\labelwidth+\labelsep+#3}}%
231     \item[Description Preamble] #2%
232   }{\end{list}\ignorespacesafterend}

233 \begin{document}
234 \begin{udescription}[Description Postamble]{uuu}(4cm)
235   \item[Item 1] xxx
236   \item[Item 2] yyy
237   \item[Description Postamble] The End
238 \end{udescription}
239 \end{document}
```

Example

```
240 \def\firstmacro{}
241 \renewkeycmd*\firstmacro<skv@>(name=Steve,
242   module=Martian logic,pass=true)[2][\@ptsize]{%
243   \edef\x{\skv@name}%
244   \wlog{\if0#1 10pt\else\if1#1 11pt\else
245     \if2#1 12pt\fi\fi\fi space font used}%
246   \def\y{#2}%
247 }

248 \firstmacro[0]{aaa}(name=John,module=Philosophy,pass=false)
```

Example

```
249 \newtwoptcmd*\macro[3][xxx]{\def\x{#1}\def\y{#2}\def\z{#3}}
250 \macro[uuu]{vvv}(www)
251 \macro{vvv}(www)
252 \macro{vvv}
```

Example

```

253 \SKV@undef\macro
254 \newtwoptcmd*\macro[2][xxx]{\def\x{#1}\def\y{#2}}
255 \macro[uuu](vvv)
256 \macro(vvv)

```

Example

```

257 \renewtwoptcmd\macro[2][xxx]{\def\x{#1}\long\def\y{#2}}
258 \macro[uuu](\par)
259 \macro(\par)

```

Example

```

260 \let\tcl\textcolor
261 \newkeycmd*\firstrule(raise=.5ex,width=1em,thick=2pt,
262   proclaim=false)[1]{%
263   \tcl{blue}{\rule[\fir@raise]{\fir@width}{\fir@thick}}%
264   #1%
265   \tcl{cyan}{\rule[\fir@raise]{\fir@width}{\fir@thick}}%
266   \iffir@proclaim \color{red}\fi\textdaggerdbl
267 }

268 \usepackage[left=2cm,right=2cm]{geometry}
269 \begin{document}
270 \parindent\z@
271 \begin{tabular*}{\textwidth}{lr}
272 \verb+\firstrule{Hello World}(width=2em,thick=4pt,
273   proclaim)+:&
274   \firstrule{Hello World}(width=2em,thick=4pt)\cr
275 \verb+\firstrule{Hello}(width=2em,thick=.5pt,
276   proclaim=true)+:&
277   \firstrule{Hello}(width=2em,thick=.5pt,proclaim=true)\cr
278 \verb+\firstrule{Hello World}(thick=2pt,
279   proclaim=true)+:&
280   \firstrule{Hello World}(thick=2pt,proclaim=true)\cr
281 \verb+\firstrule{Hello World}(raise=1ex,width=2em,
282   thick=1pt)+:&
283   \firstrule{Hello}(raise=1ex,width=2em,thick=1pt)
284 \end{tabular*}
285 \end{document}

```

Example

```

286 \let\tcl\textcolor
287 \newkeycmd\secondrule<mp@>(raise=.5ex,width=1em,thick=2pt,
288   proclaim=false)[2][\tcl{magenta}{\star$}]{%
289   \tcl{cyan}{\rule[\mp@raise]{\mp@width}{\mp@thick}}%
290   #1#2#1%
291   \tcl{blue}{\rule[\mp@raise]{\mp@width}{\mp@thick}}%
292   \ifmp@proclaim \color{red}\fi\textdaggerdbl
293 }

294 \usepackage[left=2cm,right=2cm]{geometry}
295 \begin{document}
296 \parindent\z@
297 \begin{tabular*}{\textwidth}{lr}
298 \verb+\secondrule[\textbullet]{Hello World}(width=2em,

```

```
299     thick=4pt,proclaim)+:&
300     \secondrule[\textbullet]{Hello World}(width=2em,
301       thick=4pt)\cr
302 \verb+\secondrule{Hello}(width=2em,thick=.5pt,
303   proclaim=true)+:&
304     \secondrule{Hello}(width=2em,thick=.5pt,proclaim=true)\cr
305 \verb+\secondrule{Hello World}(thick=2pt,
306   proclaim=true)+:&
307     \secondrule{Hello World}(thick=2pt,proclaim=true)\cr
308 \verb+\secondrule{Hello World}(raise=1ex,width=2em,
309   thick=1pt)+:&
310     \secondrule{Hello}(raise=1ex,width=2em,thick=1pt)
311 \end{tabular*}
312 \end{document}
```