

# The docext Package<sup>\*</sup>

Mingyu Xia<sup>†</sup>

Released 2026-06-07 v0.1A

## Abstract

The docext package is an extension for documenting the  $\text{\LaTeX}$  source files. It is based on the doc package and works smoothly under the l3doc, ltxdoc, and article classes.

## 1 Loading the Package

Write the line in the preamble

```
\usepackage[options]{docext}
```

while the package  $\langle options \rangle$  accepts the following keys

`color` =  $\langle l3color | HTML \rangle$ : The color of keys (Default: black).

`font` =  $\langle font macros \rangle$ : Font of the keys.

`delimiter` =  $\langle string \rangle$ : Delimiter between the keys (Default: , \, ).

`ratio` =  $\langle fp num \rangle$ : Ratio of the boxes' widths in codemo environment (Default: 0.625).

## 2 User's Interfaces

---

```
\docextset \docextset {\langle options \rangle}
```

The `\docextset` can be used everywhere, and only the macros used in this package *after* it will be influenced. The  $\langle options \rangle$  accepts the same keys as the package  $\langle options \rangle$  referred in [section 1](#).

---

<sup>\*</sup><https://ctan.org/pkg/docext>, <https://github.com/myhsia/docext>

<sup>†</sup>[xiamingyu@westlake.edu.cn](mailto:xiamingyu@westlake.edu.cn)

---

`\keys` `\keys [<options>] {<comma list>}`

---

Typeset the keys with specific format. The `<options>` accepts all the keys referred in [section 1](#), and also accepts the key `emph` to assign # of the keys will be emphasised.

<pre>\keys [font = \sl] {keya, keyb, keyc}\ \keys [emph = 2, delimiter = \textbar] {true, false, not given}</pre>	<pre>keya, keyb, keyc true false not given</pre>
---	--

---

`\TF` Shortcut for typeset Boolean choices.

`\TTF`

`\TFF`

---

<code>\TF, \TTF, \TFF.</code>	<code>true false, true false, true false.</code>
-------------------------------	--

`codemo`, `codemo* (env.)` Typeset the code on the left hand side and then execute the code in the right hand side, while the star version will put the code above the demo. The effect has been demoed in the introduction of user's interfaces above.

```
\begin{codemo(*)}  
Code needs to be typeset in verbatim and demoed  
\end{codemo(*)}
```

`keyval (env.)` Itemize-type environment for typesetting the keys. The effect is the same as the package options introduction of [section 1](#) in this manual.

```
\begin{keyval}  
\item [\keys{keya, keyb, keyc}]\meta{foo}  
\end{keyval}
```

## A The source code

Using codedoc as the namespace.

```
1 <@@=codedoc>
```

Start the optionlist package for l3docstrip.

```
2 <*package>
```

```
__color_named_docolor_prop
\g__codedoc_keys_font_tl
\g__codedoc_keys_delimiter_tl
g__codedoc_codemo_ratio_fp
```

Key-value definitions of the package options, separate the keys into two groups. The first one, @@ / pkgopt / keys, is also used in \keys.

```
3 \keys_define:nn { __codedoc / pkgopt / keys }
4 {
5   color      .code:n =
6   {
7     \color_if_exist:nTF {#1}
8     { \color_set:nn { docolor } {#1} }
9     { \color_set:nnn { docolor } { HTML } {#1} }
10  },
11   color      .initial:n = { black },
12   font       .tl_set:N = \g__codedoc_keys_font_tl,
13   delimiter  .tl_set:N = \g__codedoc_keys_delimiter_tl,
14   delimiter  .initial:n = { ,\, },
15 }
16 \keys_define:nn { __codedoc / pkgopt / codemo }
17 {
18   ratio      .fp_set:N = \g__codedoc_codemo_ratio_fp,
19   ratio      .initial:n = { .625 },
20 }
```

*(End of definition for \l\_\_color\_named\_docolor\_prop and others.)*

Process the class options.

```
21 \ProcessKeyOptions [ __codedoc / pkgopt / keys,
22                   __codedoc / pkgopt / codemo ]
```

**\docextset** To control the variables.

```
23 \DeclareDocumentCommand \docextset { m }
24 {
25   \keys_set_known:nn { __codedoc / pkgopt / keys } {#1}
26   \keys_set_known:nn { __codedoc / pkgopt / codemo } {#1}
27 }
```

*(End of definition for \docextset. This function is documented on page 1.)*

Load the dependencies.

```
28 \RequirePackage { xcolor, enumitem, verbatim }
```

Overwrite `\prepare@family@series@update` again under `pLATEX` and `upLATEX` to resolve the error raised by `\settowidth \MacroIndent` in the `doc` package.

```

29 \bool_lazy_or:nnT { \sys_if_engine_ptex_p: } { \sys_if_engine_uptex_p: }
30 {
31   \def\prepare@family@series@update#1#2{%
32     \def\@currentmetafamily{#1}%
33     \if@forced@series
34       \fontfamily#2%
35     \else
36       \expand@font@defaults
37       \let\target@series@value\@empty
38       \expandafter\edef\csname ??def@ult\endcsname{\f@family}%
39       \let\@elt@update@series@target@value
40         \@meta@family@list
41         \@elt{??}%
42       \let\@elt\relax
43       \fontfamily#2%
44       \ifx\target@series@value\@empty
45       \else
46         \ifx \f@series\target@series@value
47         \else
48           \maybe@load@fontshape
49           \series@maybe@drop@one@m\target@series@value\f@series
50         \fi
51       \fi
52     \fi
53   }
54 }

```

Patch the font of the macrocode environment.

```

55 \hook_gput_code:nnn { cmd / macro@font / after } { . } { \ttfamily }

```

Patch the macro `\meta` in the `l3doc` class and the `doc` package. Since the `l3doc` package has already loaded the `doc` package, a branch is required.

```

56 \cs_if_exist:cTF { ver@l3doc.cls }

```

Under the `l3doc` class, surround `\hbox:n` to the argument of the kernel function `\__codedoc_meta_original:n` of the `\meta` macro. Suppress the warning raised by `l3doc` (checking `l3syntax`) BTW.

```

57 {
58   \cs_set_eq:NN \__codedoc_meta_patch:n \__codedoc_meta_original:n
59   \cs_set_protected:Npn \__codedoc_meta_original:n #1
60     { \__codedoc_meta_patch:n { \hbox:n {#1} } }
61   \msg_redirect_module:nnn { l3doc } { warning } { none }
62 }

```

Under other classes, load the doc package then take the same trick as in the l3doc class.

```

63 {
64   \RequirePackage { doc }
65   \cs_set_eq:NN \__codedoc_meta_patch:n \meta
66   \DeclareDocumentCommand \meta { m }
67     { \__codedoc_meta_patch:n { \hbox:n {#1} } }
68 }

```

**\keys**  $\LaTeX$  2 $\epsilon$  user's interface for input keys quickly.

```

69 \DeclareDocumentCommand \keys { 0 {} m }
70 {
71   \group_begin:
72     \keys_set_known:nn { __codedoc / pkgopt / keys } {#1}
73     \keys_set_known:nn { __codedoc / keys } {#1}
74     \__codedoc_keys_aux:nnnnn
75       {#2} { \l__codedoc_keys_emph_int }
76             { \g__codedoc_keys_delimiter_tl }
77             { \g__codedoc_keys_font_tl \ttfamily }
78             { docolor }
79     \tl_if_eq:NnT \@currenvar { keyval }
80       { \g__codedoc_keys_font_tl \ttfamily \, = \, }
81   \group_end:
82 }

```

Key-value definitions special for the \keys macro.

```

\l__codedoc_keys_emph_int 83 \keys_define:nn { __codedoc / keys }
84 { emph .int_set:N = \l__codedoc_keys_emph_int }

```

(End of definition for \keys and \l\_\_codedoc\_keys\_emph\_int. This function is documented on page 2.)

\\_\_codedoc\_keys\_aux:nnnnn Auxiliary commands of the \keys macro.

\\_\_codedoc\_keys\_emph:n

**#1**: comma list, **#4**: font,  
**#2**: # of the element to be emphasised, **#5**: color (works only for the key),  
**#3**: delimiter,

```

85 \cs_new_protected_nopar:Npn \__codedoc_keys_aux:nnnnn #1#2#3#4#5
86 {
87   \group_begin:
88     \seq_set_from_clist:Nn \l_tmpa_seq {#1}
89     \int_if_zero:nF {#2}
90     {
91       \seq_gset_item:Nnn \l_tmpa_seq {#2}
92       { \__codedoc_keys_emph:n { \clist_item:nn {#1} {#2} } }

```

```

93     }
94     \seq_set_map:Nn \l_tmpb_seq \l_tmpa_seq
95     {
96         \color_group_begin:
97         \exp_args:No \color_select:n {#5} ##1
98         \color_group_end:
99     }
100     #4 \seq_use:Nn \l_tmpb_seq {#3}
101 \group_end:
102 }
103 \cs_new_protected_nopar:Npn \__codedoc_keys_emph:n #1
104 {
105     \group_begin:
106     \upshape \bfseries \exp_args:No \color_select:n { red } #1
107     \group_end:
108 }

```

(End of definition for `\__codedoc_keys_aux:nnnnn` and `\__codedoc_keys_emph:n`.)

**\TF** L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> user's interface for input Boolean choices quickly.

**\TTF** `\DeclareDocumentCommand \TF { } { \__codedoc_keys_TF_aux:n { 0 } }`

**\TFF** `\DeclareDocumentCommand \TTF { } { \__codedoc_keys_TF_aux:n { 1 } }`

`\DeclareDocumentCommand \TFF { } { \__codedoc_keys_TF_aux:n { 2 } }`

(End of definition for `\TF`, `\TTF`, and `\TFF`. These functions are documented on page 2.)

`\__codedoc_keys_TF_aux:n` Auxiliary commands of the `\TF`, `\TTF`, `\TFF` macros.

```

112 \cs_new_protected_nopar:Npn \__codedoc_keys_TF_aux:n #1
113 {
114     \__codedoc_keys_aux:nnnnn
115     { \g__codedoc_true_tl, \g__codedoc_false_tl } {#1}
116     { \textup \g__codedoc_vbar_tl } { \ttfamily } { docolor }
117 }

```

Constant variables used in `\__codedoc_keys_TF_aux:n`.

```

\g__codedoc_true_tl 118 \tl_const:Nn \g__codedoc_true_tl { true }
\g__codedoc_false_tl 119 \tl_const:Nn \g__codedoc_false_tl { false }
\g__codedoc_vbar_tl 120 \tl_const:Nn \g__codedoc_vbar_tl
121 { \sys_if_engine_opentype:TF { \textbar } { | } }

```

(End of definition for `\__codedoc_keys_TF_aux:n` and others.)

`keyval (env)`

```

122 \newlist{keyval}{itemize}{10}
123 \setlist[keyval]{leftmargin = 0pt, labelsep = 0pt}

```

codemo, codemo\* (*env*) Environment for typesetting the code within a verbatim and a demo environment at the same time.

```

124 \DeclareDocumentEnvironment { codemo } { } { \_codedoc_codemo_start: }
125 {
126   \_codedoc_codemo_end:Nn \c_false_bool
127   { \g\_codedoc_codemo_ratio_fp }
128 }
129 \DeclareDocumentEnvironment { codemo* } { } { \_codedoc_codemo_start: }
130 {
131   \_codedoc_codemo_end:Nn \c_true_bool
132   { \g\_codedoc_codemo_ratio_fp }
133 }

```

\\_codedoc\_codemo\_start: Auxiliary commands of the codemo environment.

\\_codedoc\_codemo\_end:Nn 134 \cs\_new\_protected:Npn \\_codedoc\_codemo\_start:

135 {  
The I/O streams part<sup>1</sup>.

```

136 \group_begin:
137   \@bsphack
138   \iow_open:Nn \l\_codedoc_codemo_iow { \l\_codedoc_codemo_name_tl }
139   \let \do \char_set_catcode_other:N
140   \dospecials \char_set_catcode_active:N \^M
141   \cs_set:Npn \verbatim@processline
142   {
143     \exp_args:NNo \iow_now:Nn
144     \l\_codedoc_codemo_iow { \int_use:N \verbatim@line }
145   }
146   \verbatim@start
147 }
148 \cs_new_protected:Npn \_codedoc_codemo_end:Nn #1#2
149 {
150   \iow_close:N \l\_codedoc_codemo_iow
151   \@esphack
152   \group_end:
153   \char_set_catcode_comment:N \%
154   \ior_open:Nn \l\_codedoc_codemo_ior { \l\_codedoc_codemo_name_tl }
155   \int_zero:N \l\_codedoc_codemo_int
156   \ior_map_inline:Nn \l\_codedoc_codemo_ior
157   { \int_incr:N \l\_codedoc_codemo_int }

```

The typeset part.

```

158 \par \smallskip

```

---

<sup>1</sup>By modifying @AlphaZTX's [phy2docdef.tex](#) magically using LaTeX3.

```

159 \group_begin:
160   \dim_set:Nn \parindent { \c_zero_skip }
161   \fbox
162   {
163     \bool_if:NT #1 { \null \skip_horizontal:n { 2ex } }
164     \parbox[c][\l__codedoc_codemo_int \baselineskip + 1ex]
165     {
166       \fp_eval:n { \bool_if:NTF #1 { 1 } { #2 } * .96 }
167       \linewidth \bool_if:NT #1 { -.8pt }
168     } { \small \verbatiminput \l__codedoc_codemo_name_tl }
169   }
170   \bool_if:NTF #1
171   { \par \skip_vertical:n { -1.5pt } } { \hfill }
172   \fbox
173   {
174     \null \skip_horizontal:n { 2ex }
175     \parbox[c][\l__codedoc_codemo_int \baselineskip + 1ex]
176     {
177       \fp_eval:n { \bool_if:NTF #1 { 1 } { (1 - #2) } * .96 }
178       \linewidth \bool_if:NTF #1 { -.8pt } { - 2ex }
179     } { \file_input:n { \l__codedoc_codemo_name_tl } }
180   }
181   \par \smallskip
182 \group_end:
183 \char_set_catcode_ignore:N \%
184 }

```

Variables used in \l\_\_codedoc\_codemo\_start: and \l\_\_codedoc\_codemo\_end:Nn.

```

185 \l__codedoc_codemo_iow \l__codedoc_codemo_iow
186 \l__codedoc_codemo_ior \l__codedoc_codemo_ior
187 \l__codedoc_codemo_int \l__codedoc_codemo_int
188 \l__codedoc_codemo_name_tl \l__codedoc_codemo_name_tl {\jobname.codemo.vrb}
\l__codedoc_codemo_name_tl

```

*(End of definition for \l\_\_codedoc\_codemo\_start: and others.)*

End the optionlist package for l3docstrip.

```

189 \</package>

```

Restore the namespace.

```

190 \<@@=

```



# Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
\%	153, 183
\,	14, 80
\^	140
B	
\baselineskip	164, 175
\bfseries	106
bool commands:	
\bool_if:NTF	163, 166, 167, 170, 177, 178
\bool_lazy_or:nnTF	29
\c_false_bool	126
\c_true_bool	131
C	
char commands:	
\char_set_catcode_active:N	140
\char_set_catcode_comment:N	153
\char_set_catcode_ignore:N	183
\char_set_catcode_other:N	139
clist commands:	
\clist_item:nn	92
codedoc internal commands:	
\__codedoc_codemo_end:Nn	8, 126, 131, <u>134</u> , 148
\l__codedoc_codemo_int	155, 157, 164, 175, <u>185</u>
\l__codedoc_codemo_ior	154, 156, <u>185</u>
\l__codedoc_codemo_iow	138, 144, 150, <u>185</u>
\l__codedoc_codemo_name_tl	138, 154, 168, 179, <u>185</u>
\g__codedoc_codemo_ratio_fp	3, 127, 132
\__codedoc_codemo_start:	8, 124, 129, <u>134</u> , 134
\g__codedoc_false_tl	115, <u>118</u>
\__codedoc_keys_aux:nnnnn	74, <u>85</u> , 85, 114
\g__codedoc_keys_delimiter_tl	3, 76
\__codedoc_keys_emph:n	<u>85</u> , 92, 103
\l__codedoc_keys_emph_int	75, <u>83</u>
\g__codedoc_keys_font_tl	3, 77, 80
\__codedoc_keys_TF_aux:n	6, 109, 110, 111, <u>112</u> , 112
\__codedoc_meta_original:n	4, 58, 59
\__codedoc_meta_patch:n	58, 60, 65, 67
\g__codedoc_true_tl	115, <u>118</u>
\g__codedoc_vbar_tl	116, <u>118</u>
codemo, codemo* (env.)	2, 7
color commands:	
\color_group_begin:	96
\color_group_end:	98
\color_if_exist:nTF	7
\color_select:n	97, 106
\color_set:nn	8
\color_set:nnn	9
color internal commands:	
\l_color_named_docolour_prop	3
cs commands:	
\cs_if_exist:NTF	56
\cs_new_protected:Npn	134, 148
\cs_new_protected_nopar:Npn	85, 103, 112
\cs_set:Npn	141
\cs_set_eq:NN	58, 65
\cs_set_protected:Npn	59
\csname	38
D	
\DeclareDocumentCommand	23, 66, 69, 109, 110, 111
\DeclareDocumentEnvironment	124, 129
\def	31, 32

dim commands:	\int_zero:N . . . . . 155
\dim_set:Nn . . . . . 160	ior commands:
\do . . . . . 139	\ior_map_inline:Nn . . . . . 156
\docextset . . . . . 1, 23	\ior_new:N . . . . . 186
\dospecials . . . . . 140	\ior_open:Nn . . . . . 154
<b>E</b>	iow commands:
\edef . . . . . 38	\iow_close:N . . . . . 150
\else . . . . . 35, 45, 47	\iow_new:N . . . . . 185
\endcsname . . . . . 38	\iow_now:Nn . . . . . 143
environments:	\iow_open:Nn . . . . . 138
codemo, codemo* . . . . . 2, 7	<b>J</b>
keyval . . . . . 2, 6	\jobname . . . . . 188
exp commands:	<b>K</b>
\exp_args:NNo . . . . . 143	\keys . . . . . 2, 69
\exp_args:No . . . . . 97, 106	keys commands:
\expandafter . . . . . 38	\keys_define:nn . . . . . 3, 16, 83
<b>F</b>	\keys_set_known:nn . . . 25, 26, 72, 73
\fbox . . . . . 161, 172	keyval (env.) . . . . . 2, 6
\fi . . . . . 50, 51, 52	<b>L</b>
file commands:	\let . . . . . 37, 39, 42, 139
\file_input:n . . . . . 179	\linewidth . . . . . 167, 178
\fontfamily . . . . . 34, 43	<b>M</b>
fp commands:	\meta . . . . . 65, 66
\fp_eval:n . . . . . 166, 177	msg commands:
<b>G</b>	\msg_redirect_module:nnn . . . . . 61
group commands:	<b>N</b>
\group_begin: . . . 71, 87, 105, 136, 159	\newlist . . . . . 122
\group_end: . . . 81, 101, 107, 152, 182	\null . . . . . 163, 174
<b>H</b>	<b>P</b>
hbox commands:	\par . . . . . 158, 171, 181
\hbox:n . . . . . 4, 60, 67	\parbox . . . . . 164, 175
\hfill . . . . . 171	\parindent . . . . . 160
hook commands:	\ProcessKeyOptions . . . . . 21
\hook_gput_code:nnn . . . . . 55	<b>R</b>
<b>I</b>	\relax . . . . . 42
\ifx . . . . . 44, 46	\RequirePackage . . . . . 28, 64
int commands:	<b>S</b>
\int_if_zero:nTF . . . . . 89	seq commands:
\int_incr:N . . . . . 157	\seq_gset_item:Nnn . . . . . 91
\int_new:N . . . . . 187	
\int_use:N . . . . . 144	

