

The docext Package^{*}

Mingyu Xia[†]

Released 2026-06-13 v0.2A

Abstract

The docext package is an extension for documenting the L^AT_EX source files. It is based on the doc package and works smoothly under the l₃doc, ltxdoc, and article classes.

1 Loading & Configuring the Package

Write the line in the preamble

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

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

`color` = $\langle 13color | HTML \rangle$: The color of keys (Default: black).

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

`delimiter` = $\langle string \rangle$: Delimiter between the keys.

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

Apart from configuring the package with the package options, users can also use the `\docextset` function within the preamble and the context.

<code>\docextset</code>	<code>\docextset {\langle options \rangle}</code>
-------------------------	---

New: 2026-06-07	The functions used in this package <i>after</i> <code>\docextset</code> will be influenced. The $\langle options \rangle$ accepts the same keys above.
-----------------	---

^{*}<https://ctan.org/pkg/docext>, <https://github.com/myhsia/docext>

[†]xiamingyu@westlake.edu.cn

2 User's Interfaces

<code>\keys</code>	<code>\keys [<options>] {<comma list>}</code>
New: 2026-06-07 Updated: 2026-06-13	Typeset the keys with specific format. The <code><options></code> accepts the <code>delimiter</code> , <code>font</code> , <code>color</code> keys referred in section 1 and another <code>emph</code> key to assign # of the key will be emphasised.

The initial value of the `delimiter` key is `,\,\allowbreak` in this function.

```
\keys [font = \sl] {keya, keyb, keyc}\\
\keys [emph = 2, delimiter = \textbar]
{true, false, not given}
```

`keya`, `keyb`, `keyc`
`true`|`false`|`not given`

<code>\val</code>	<code>\val [<options>] {<comma list>}</code>
New: 2026-06-13	Typeset the value with specific format and surround with <code>\meta</code> . The <code><options></code> accepts the <code>delimiter</code> , <code>font</code> , <code>color</code> keys referred in section 1 . The initial value of the <code>delimiter</code> key is <code>\textbar</code> in this function.

<code>\TF</code>	Shortcut for typeset Boolean-choice value.
------------------	--

New: 2026-06-07	<code>\TF</code> , <code>\TTF</code> , <code>\TFF</code> .	<code>true</code> <code>false</code> , <code>true</code> <code>false</code> , <code>true</code> <code>false</code> .
-----------------	--	--

<code>\ttwd</code>	<code>\hspace \<fontsize>ttwd</code>
--------------------	--

New: 2026-06-13	Output the dimension of the width of one typewriter letter. The <code><fontsize></code> prefix can be chosen as <code>tiny</code> , <code>scriptsize</code> , <code>footnotesize</code> , <code>small</code> , <code>normalsize</code> , <code>large</code> , <code>Large</code> , <code>LARGE</code> , <code>huge</code> , <code>Huge</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.

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

`keyval (env)` The effect is the same as the demos in the introductions of the functions above. Itemize-type environment for typesetting the keys.

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

The effect is the same as the package options introduction of [section 1](#) in this manual.

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  }
15 \keys_define:nn { __codedoc / pkgopt / codemo }
16 {
17   ratio      .fp_set:N = \g__codedoc_codemo_ratio_fp,
18   ratio      .initial:n = { .625 },
19  }
```

(End of definition for \l__color_named_docolor_prop and others.)

Process the class options.

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

\docextset To control the variables.

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

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

Load the dependencies.

```
27 \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.

```

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

```

Patch the font of the macrocode environment.

```

54 \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.

```

55 \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`), and reset the counter: section at the beginning of the documentation environment.

```

56 {
57   \cs_set_eq:NN \__codedoc_meta_patch:n \__codedoc_meta_original:n
58   \cs_set_protected:Npn \__codedoc_meta_original:n #1
59     { \__codedoc_meta_patch:n { \hbox:n {#1} } }
60   \msg_redirect_module:nnn { l3doc } { warning } { none }

```

```

61 \hook_gput_code:nnn { env / documentation / before } { . }
62 { \setcounter { section } { 0 } }
63 }

```

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

```

64 {
65 \RequirePackage { doc }
66 \cs_set_eq:NN \__codedoc_meta_patch:n \meta
67 \DeclareDocumentCommand \meta { m }
68 { \__codedoc_meta_patch:n { \hbox:n {#1} } }
69 }

```

\keys L^AT_EX 2_ε user's interface for input keys quickly.

```

70 \DeclareDocumentCommand \keys { 0 {} m }
71 {
72 \group_begin:
73 \keys_set_known:nn { __codedoc / pkgopt / keys }
74 { delimiter = { ,\,\allowbreak }, #1 }
75 \keys_set_known:nn { __codedoc / keys } {#1}
76 \__codedoc_keys_aux:nnnnn {#2}
77 { \l__codedoc_keys_emph_int }
78 { \textup \g__codedoc_keys_delimiter_tl }
79 { \g__codedoc_keys_font_tl \ttfamily }
80 { docolor }
81 \tl_if_eq:NnT \@currenvir { keyval }
82 { \g__codedoc_keys_font_tl \ttfamily \, = \, }
83 \group_end:
84 }

```

Key-value definitions special for the \keys macro.

```

\l__codedoc_keys_emph_int 85 \keys_define:nn { __codedoc / keys }
86 { 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.)

\val L^AT_EX 2_ε user's interface for input value quickly.

```

87 \DeclareDocumentCommand \val { 0 {} m }
88 {
89 \group_begin:
90 \keys_set_known:nn { __codedoc / pkgopt / keys }
91 { delimiter = { \g__codedoc_vbar_tl \allowbreak }, #1 }
92 \meta
93 {
94 \__codedoc_keys_aux:nnnnn {#2}
95 { 0 }

```

```

96         { \textup \g__codedoc_keys_delimiter_tl }
97         { \g__codedoc_keys_font_tl \ttfamily      }
98         { docolor                                }
99     }
100 \group_end:
101 }

```

(End of definition for \val. 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,

```

102 \cs_new_protected_nopar:Npn \__codedoc_keys_aux:nnnnn #1#2#3#4#5
103 {
104   \group_begin:
105   \seq_set_from_clist:Nn \l_tmpa_seq {#1}
106   \int_if_zero:nF {#2}
107   {
108     \seq_gset_item:Nnn \l_tmpa_seq {#2}
109     { \__codedoc_keys_emph:n { \clist_item:nn {#1} {#2} } }
110   }
111   \seq_set_map:Nnn \l_tmpb_seq \l_tmpa_seq
112   {
113     \color_group_begin:
114     \exp_args:No \color_select:n {#5} ##1
115     \color_group_end:
116   }
117   #4 \seq_use:Nn \l_tmpb_seq {#3}
118   \group_end:
119 }
120 \cs_new_protected_nopar:Npn \__codedoc_keys_emph:n #1
121 {
122   \group_begin:
123   \upshape \bfseries \exp_args:No \color_select:n { red } #1
124   \group_end:
125 }

```

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

\TF $\LaTeX 2_{\epsilon}$ user's interface for input Boolean-choice value quickly.

```

\TTF 126 \DeclareDocumentCommand \TF {} { \__codedoc_keys_TF_aux:n { 0 } }
\TTF 127 \DeclareDocumentCommand \TTF {} { \__codedoc_keys_TF_aux:n { 1 } }
\TTF 128 \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.
129 \cs_new_protected_nopar:Npn \__codedoc_keys_TF_aux:n #1
130 {
131   \__codedoc_keys_aux:nnnnn
132   { \g__codedoc_true_tl, \g__codedoc_false_tl } {#1}
133   { \textup \g__codedoc_vbar_tl } { \ttfamily } { docolor }
134 }
```

Constant variables used in `__codedoc_keys_TF_aux:n`.

```
\g__codedoc_true_tl 135 \tl_const:Nn \g__codedoc_true_tl { true }
\g__codedoc_false_tl 136 \tl_const:Nn \g__codedoc_false_tl { false }
\g__codedoc_vbar_tl 137 \tl_const:Nn \g__codedoc_vbar_tl
138 { \sys_if_engine_opentype:TF { \textbar } { | } }
```

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

keyval (*env*) Itemize-type environment for typesetting the keys. Using the interfaces provided by the `enumitem` package.

```
139 \newlist{keyval}{itemize}{10}
140 \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.

```
141 \DeclareDocumentEnvironment { codemo } { } { \__codedoc_codemo_start: }
142 {
143   \__codedoc_codemo_end:Nn \c_false_bool
144   { \g__codedoc_codemo_ratio_fp }
145 }
146 \DeclareDocumentEnvironment { codemo* } { } { \__codedoc_codemo_start: }
147 {
148   \__codedoc_codemo_end:Nn \c_true_bool
149   { \g__codedoc_codemo_ratio_fp }
150 }
```

`__codedoc_codemo_start:` Auxiliary commands of the `codemo` environment.

```
\__codedoc_codemo_end:Nn 151 \cs_new_protected:Npn \__codedoc_codemo_start:
152 {
```

The I/O streams part¹.

```
153 \group_begin:
154 \@bsphack
155 \iow_open:Nn \l__codedoc_codemo_iow { \l__codedoc_codemo_name_tl }
```

¹By modifying `@AlphaZTX's phy2docdef.tex` magically using LaTeX3.

```

156 \let \do \char_set_catcode_other:N
157 \dospecials \char_set_catcode_active:N \^^M
158 \cs_set:Npn \verbatim@processline
159 {
160   \exp_args:NNo \iow_now:Nn
161   \l__codedoc_codemo_iow { \int_use:N \verbatim@line }
162 }
163 \verbatim@start
164 }
165 \cs_new_protected:Npn \__codedoc_codemo_end:Nn #1#2
166 {
167   \iow_close:N \l__codedoc_codemo_iow
168   \@esphack
169   \group_end:
170   \char_set_catcode_comment:N \%
171   \ior_open:Nn \l__codedoc_codemo_ior { \l__codedoc_codemo_name_tl }
172   \int_zero:N \l__codedoc_codemo_int
173   \ior_map_inline:Nn \l__codedoc_codemo_ior
174   { \int_incr:N \l__codedoc_codemo_int }

```

The typeset part.

```

175 \par \smallskip
176 \group_begin:
177 \dim_set:Nn \parindent { \c_zero_skip }
178 \fbox
179 {
180   \bool_if:NT #1 { \null \skip_horizontal:n { 2\ttwd } }
181   \parbox [ c ] [ \l__codedoc_codemo_int \baselineskip + .48ex ]
182   {
183     \fp_eval:n { \bool_if:NTF #1 { 1 } { #2 } * .96 }
184     \linewidth \bool_if:NT #1 { -.8pt }
185     } { \small \verbatiminput \l__codedoc_codemo_name_tl }
186   }
187   \bool_if:NTF #1
188   { \par \skip_vertical:n { -1.5pt } } { \hfill }
189   \fbox
190   {
191     \null \skip_horizontal:n { 2\ttwd }
192     \parbox [ c ] [ \l__codedoc_codemo_int \baselineskip + .48ex ]
193     {
194       \fp_eval:n { \bool_if:NTF #1 { 1 } { (1 - #2) } * .96 }
195       \linewidth \bool_if:NTF #1 { -.8pt } { - 2\ttwd }
196       } { \file_input:n { \l__codedoc_codemo_name_tl } }
197     }
198   \par \smallskip

```



```

199   \group_end:
200   \char_set_catcode_ignore:N \%
201 }

```

Variables used in \l__codedoc_codemo_start: and \l__codedoc_codemo_end:Nn.

```

202 \iow_new:N    \l__codedoc_codemo_iow
\l__codedoc_codemo_iow 203 \ior_new:N    \l__codedoc_codemo_ior
\l__codedoc_codemo_ior 204 \int_new:N    \l__codedoc_codemo_int
\l__codedoc_codemo_int 205 \tl_const:Nn \l__codedoc_codemo_name_tl {\jobname.codemo.vrb}
\l__codedoc_codemo_name_tl

```

(End of definition for \l__codedoc_codemo_start: and others.)

\ttwd Dimensions of the width of one typewriter letter under different font sizes.

Box for saving one typewriter letter temporally.

```

\l__codedoc_ttwd_box 206 \box_new:N \l__codedoc_ttwd_box
207 \clist_map_inline:nn
208 {
209   tiny, scriptsize, footnotesize, small, \@empty,
210   large, Large,          LARGE,          huge, Huge
211 }
212 {
213   \dim_new:c { #1 ttwd }
214   \cs_new_protected_nopar:cpn { update#1ttwd }
215   {
216     \hbox_set:Nn \l__codedoc_ttwd_box { \use:c {#1} \ttfamily x }
217     \dim_gset:cn { #1 ttwd } { \box_wd:N \l__codedoc_ttwd_box }
218   }
219   \hook_gput_code:nnn { begindocument / before } { . }
220   { \use:c { update#1ttwd } }
221 }

```

(End of definition for \ttwd and \l__codedoc_ttwd_box. This function is documented on page 2.)

End the optionlist package for l3docstrip.

```

222 \</package>

```

Restore the namespace.

```

223 \<@=@>

```

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	
<code>\%</code>	170, 200
<code>\,</code>	74, 82
<code>\(fontsize)ttwd</code>	2
<code>\^</code>	157
A	
<code>\allowbreak</code>	74, 91
B	
<code>\baselineskip</code>	181, 192
<code>\bfseries</code>	123
bool commands:	
<code>\bool_if:NTF</code>	180, 183, 184, 187, 194, 195
<code>\bool_lazy_or:nnTF</code>	28
<code>\c_false_bool</code>	143
<code>\c_true_bool</code>	148
box commands:	
<code>\box_new:N</code>	206
<code>\box_wd:N</code>	217
C	
char commands:	
<code>\char_set_catcode_active:N</code> ..	157
<code>\char_set_catcode_comment:N</code> ..	170
<code>\char_set_catcode_ignore:N</code> ..	200
<code>\char_set_catcode_other:N</code> ...	156
clist commands:	
<code>\clist_item:nn</code>	109
<code>\clist_map_inline:nn</code>	207
codedoc internal commands:	
<code>__codedoc_codemo_end:Nn</code>	
.....	9, 143, 148, <u>151</u> , 165
<code>\l__codedoc_codemo_int</code>	
.....	172, 174, 181, 192, <u>202</u>
<code>\l__codedoc_codemo_ior</code>	171, 173, <u>202</u>
<code>\l__codedoc_codemo_iow</code>	
.....	155, 161, 167, <u>202</u>
<code>\l__codedoc_codemo_name_tl</code> ..	
.....	155, 171, 185, 196, <u>202</u>
<code>\g__codedoc_codemo_ratio_fp</code> ..	
.....	<u>3</u> , 144, 149
<code>__codedoc_codemo_start:</code>	
.....	9, 141, 146, <u>151</u> , 151
<code>\g__codedoc_false_tl</code>	132, <u>135</u>
<code>__codedoc_keys_aux:nnnnn</code> ...	
.....	76, 94, <u>102</u> , <u>102</u> , 131
<code>\g__codedoc_keys_delimiter_tl</code>	
.....	<u>3</u> , 78, 96
<code>__codedoc_keys_emph:n</code>	<u>102</u> , 109, 120
<code>\l__codedoc_keys_emph_int</code> ..	77, <u>85</u>
<code>\g__codedoc_keys_font_tl</code>	
.....	<u>3</u> , 79, 82, 97
<code>__codedoc_keys_TF_aux:n</code>	
.....	7, 126, 127, 128, <u>129</u> , 129
<code>__codedoc_meta_original:n</code> ..	
.....	4, 57, 58
<code>__codedoc_meta_patch:n</code>	
.....	57, 59, 66, 68
<code>\g__codedoc_true_tl</code>	132, <u>135</u>
<code>\l__codedoc_ttwd_box</code> ..	206, 216, 217
<code>\g__codedoc_vbar_tl</code> ...	91, 133, <u>135</u>
codemo, codemo* (env.)	
2, 7	
color commands:	
<code>\color_group_begin:</code>	113
<code>\color_group_end:</code>	115
<code>\color_if_exist:nTF</code>	7
<code>\color_select:n</code>	114, 123
<code>\color_set:nn</code>	8
<code>\color_set:nnn</code>	9
color internal commands:	
<code>\l__color_named_docolour_prop</code> ..	<u>3</u>
cs commands:	
<code>\cs_if_exist:NTF</code>	55
<code>\cs_new_protected:Npn</code> ...	151, 165
<code>\cs_new_protected_nopar:Npn</code> ..	
.....	102, 120, 129, 214
<code>\cs_set:Npn</code>	158

P	
\par	175, 188, 198
\parbox	181, 192
\parindent	177
\ProcessKeyOptions	20
R	
\relax	41
\RequirePackage	27, 65
S	
seq commands:	
\seq_gset_item:Nnn	108
\seq_set_from_clist:Nn	105
\seq_set_map:Nnn	111
\seq_use:Nn	117
\l_tmpa_seq	105, 108, 111
\l_tmpb_seq	111, 117
\setcounter	62
\setlist	140
skip commands:	
\skip_horizontal:n	180, 191
\skip_vertical:n	188
\c_zero_skip	177
\small	185
\smallskip	175, 198
sys commands:	
\sys_if_engine_opentype:TF	138
\sys_if_engine_ptex_p:	28
\sys_if_engine_uptex_p:	28
T	
TeX and L ^A T _E X 2 _ε commands:	
\@bsphack	154
\@currentmetafamily	31
\@currenvir	81
\@elt	38, 40, 41
\@empty	36, 43, 209
\@esphack	168
\@meta@family@list	39
\docxtset	1
\expand@font@defaults	35
\fontfamily	37
\fontseries	45, 48
\if@forced@series	32
\keys	2, 3, 5, 6
\maybe@load@fontshape	47
\meta	2, 4
\prepare@family@series@update	4, 30
\series@maybe@drop@one@cm	48
\target@series@value	36, 43, 45, 48
\textbar	2
\TF	7
\TFF	7
\TTF	7
@update@series@target@value	38
\val	2
\verbatim@line	161
\verbatim@processline	158
\verbatim@start	163
\textbar	138
\textup	78, 96, 133
\TF	2, 126
\TFF	126
tl commands:	
\tl_const:Nn	135, 136, 137, 205
\tl_if_eq:NnTF	81
\TTF	126
\ttfamily	54, 79, 82, 97, 133, 216
\ttwd	2, 180, 191, 195, 206
U	
\upshape	123
use commands:	
\use:N	216, 220
V	
\val	2, 87
\verbatiminput	185