

The fancythm package

Maximilian Keßler

Released 2022-01-12

1 fancythm implementation

```
1 <*package>
2 <@@=fancythm>
```

1.1 Dependencies

```
3 \RequirePackage{l3keys2e}
4 \RequirePackage[default styles]{thmstyle}
```

1.2 Counter management

Wrappers for L^AT_EX2e counter manipulation

`\arabic:n` Obvious meanings.

```
\counter_new:n      5 \cs_set_eq:NN \arabic:n \arabic
\counter_within:nn  6 \cs_set_eq:NN \counter_new:n \newcounter
\counter_new:nn     7 \cs_set_eq:NN \counter_within:nn \counterwithin
                   8 \cs_new:Npn \counter_new:nn #1 #2
                   9 {
                  10   \counter_new:n { #1 }
                  11   \counter_within:nn { #1 } { #2 }
                  12 }
```

(End of definition for `\arabic:n` and others. These functions are documented on page ??.)

```
\counter_if_exist_p:n
\counter_if_exist:nT 13 \cs_new:Npn \counter_if_exist_p:n #1
\counter_if_exist:F  14 {
\counter_if_exist:nTF 15   \cs_if_exist_p:c { c@ #1 }
                   16 }
                   17 \cs_new:Npn \counter_if_exist:nT #1
                   18 {
                   19   \cs_if_exist:cT { c@ #1 }
                   20 }
                   21 \cs_new:Npn \counter_if_exist:nF #1
                   22 {
                   23   \cs_if_exist:cF { c@ #1 }
                   24 }
                   25 \cs_new:Npn \counter_if_exist:nTF #1
                   26 {
                   27   \cs_if_exist:cTF { c@ #1 }
                   28 }
```

(End of definition for \counter_if_exist_p:n and others. These functions are documented on page ??.)

`\counter_undefined_error:n` Standard counter error message.

```
29 \cs_set_eq:Nc \counter_undefined_error:n { @nocounterr }
```

(End of definition for \counter_undefined_error:n. This function is documented on page ??.)

`\counter_ensure_exist:n`

```
30 \cs_new:Npn \counter_ensure_exist:n #1
31 {
32   \counter_if_exist:nF { #1 }
33   {
34     \counter_undefined_error:n { #1 }
35   }
36 }
```

(End of definition for \counter_ensure_exist:n. This function is documented on page ??.)

`\counter_the:n`

`\counter_set_the:nn`

```
37 \cs_new:Npn \counter_the:n #1
38 {
39   \use:c { the #1 }
40 }
41 \cs_new:Npn \counter_set_the:nn #1 #2
42 {
43   \counter_ensure_exist:n { #1 }
44   \exp_args:Nc \renewcommand { the #1 } { #2 }
45 }
```

(End of definition for \counter_the:n and \counter_set_the:nn. These functions are documented on page ??.)

`\counter_sub:nn` `\counter_sub:nn{<counter1>}{<counter2>}`

```
46 \cs_new:Npn \counter_sub:nn #1 #2
47 {
48   \counter_within:nn { #1 } { #2 }
49   \counter_set_the:nn { #1 }
50   {
51     \counter_the:n { #2 } . \arabic:n { #1 }
52   }
53 }
```

(End of definition for \counter_sub:nn. This function is documented on page ??.)

`\counter_new_sub:nn_#1_#2`

```
54 \cs_new:Npn \counter_new_sub:nn #1 #2
55 {
56   \counter_new:n { #1 }
57   \counter_sub:nn { #1 } { #2 }
58 }
59
```

(End of definition for \counter_new_sub:nn #1 #2. This function is documented on page ??.)

1.3 Counter management

`_fancythm_counter_sub_if_exist:n`

```
60 \cs_new:Npn \_fancythm_counter_sub_if_exist:n #1
61 {
62   \counter_if_exist:nT { #1 }
63   {
64     \counter_new_sub:nn { in #1 } { #1 }
65   }
66 }
```

(End of definition for _fancythm_counter_sub_if_exist:n.)

`\counter_alias:nn`
`\counter_alias:nx`

`\counter_alias:nn{<counter1>}{<counter2>}`

Note that since 2022-06-02, the implementation of `\@counteralias` in `thmtools` has changed. Before, aliasing an already existing counter did not throw an error, which is considered a bug now (<https://github.com/muzimuzhi/thmtools/issues/28>). Since `!!` relies on the old functionality, we adapt this here by undefining `\c@<countername>` before calling `\@counteralias`.

```
67 \cs_new:Npn \counter_alias:nn #1 % trailing #2
68 {
69   \cs_undefine:c { c@ #1 }
70   \use:c { \@counteralias } { #1 } % implicit #2
71 }
72 \cs_generate_variant:Nn \counter_alias:nn { n x }
```

(End of definition for \counter_alias:nn. This function is documented on page ??.)

1.4 Initialization

`indocument`
`insection`
`insubsection`
`inchapter`
`inpart`

```
73 \counter_new:n { indocument }
74 \_fancythm_counter_sub_if_exist:n { section }
75 \_fancythm_counter_sub_if_exist:n { subsection }
76 \_fancythm_counter_sub_if_exist:n { chapter }
77 \_fancythm_counter_sub_if_exist:n { part }
```

(End of definition for indocument and others. These variables are documented on page ??.)

`toplevel`
`sublevel`
`subsublevel`
`subsubsublevel`

Note that the `toplevel` counter will be potentially redefined by the given package options.

```
78 \counter_alias:nn { toplevel } { indocument }
79 \counter_new_sub:nn { sublevel } { toplevel }
80 \counter_new_sub:nn { subsublevel } { sublevel }
81 \counter_new_sub:nn { subsubsublevel } { subsublevel }
```

(End of definition for toplevel and others. These variables are documented on page ??.)

`\g_fancythm_style_str`

This will hold the style information of the package.

```
82 \str_new:N \g_fancythm_style_str
```

(End of definition for \g_fancythm_style_str.)

`\g_fancythm_translator_bool`

```
83 \bool_new:N \g_fancythm_translator_bool
```

(End of definition for `\g__fancythm_translator_bool`.)

`\g__fancythm_generate_defaults_bool`

84 `\bool_new:N \g__fancythm_generate_defaults_bool`

(End of definition for `\g__fancythm_generate_defaults_bool`.)

`\l__fancythm_key_name_tl`

`\l__fancythm_key_mdframed_tl`

85 `\tl_new:N \l__fancythm_key_name_tl`

86 `\tl_new:N \l__fancythm_key_mdframed_tl`

`\l__fancythm_key_style_tl`

87 `\tl_new:N \l__fancythm_key_style_tl`

`\l__fancythm_key_group_clist`

88 `\clist_new:N \l__fancythm_key_group_clist`

`\l__fancythm_key_thmtools_clist`

89 `\clist_new:N \l__fancythm_key_thmtools_clist`

(End of definition for `\l__fancythm_key_name_tl` and others.)

`\l__fancythm_name_tl`

`\l__fancythm_thmtools_clist`

90 `\tl_new:N \l__fancythm_name_tl`

`\l__fancythm_group_clist`

91 `\clist_new:N \l__fancythm_thmtools_clist`

92 `\clist_new:N \l__fancythm_group_clist`

(End of definition for `\l__fancythm_name_tl`, `\l__fancythm_thmtools_clist`, and `\l__fancythm_group_clist`.)

`\g__fancythm_groupthm_option_clist`

93 `\clist_new:N \g__fancythm_groupthm_option_clist`

(End of definition for `\g__fancythm_groupthm_option_clist`.)

1.5 Key interface

TeXhackers note: Note that unfortunately, none of the keynames really contains a space. L^AT_EX2e strips spaces before loading a package, so introducing them here would make them inaccessible. Here they are ignored by L^AT_EX3 and are present for readability.

94 `\keys_define:nn { fancythm }`

95 `{`

96 `translator .bool_set:N = \g__fancythm_translator_bool ,`

97 `translator .default:n = { true } ,`

98 `generate defaults .bool_set:N = \g__fancythm_generate_defaults_bool ,`

99 `generate defaults .default:n = { true } ,`

100 `number in .choices:nn =`

101 `{ document, section, subsection, chapter, part }`

102 `{`

103 `\counter_alias:nx { toplevel } { in \tl_use:N \l_keys_choice_tl }`

104 `}`

105 `number in .default:n = { document } ,`

106 `style .choices:nn =`

107 `{ fancy, plain, classic }`

108 `{`

109 `\str_set:Nn \g__fancythm_style_str { \tl_use:N \l_keys_choice_tl }`

110 `}`

111 `style .default:n = { fancy } ,`

112 `cache .code:n =`

```

113     {
114       \clist_put_right:Nn \g__fancythm_groupthm_option_clist { cache = #1 }
115     }
116     cache .default:n = { true }
117     cache version .code:n =
118     {
119       \clist_put_right:Nn \g__fancythm_groupthm_option_clist { cache version = #1 }
120     }
121     cache version .default:n = { 0 }
122   }
123 \keys_define:nn { fancythm / fancytheorem }
124 {
125   name .tl_set:N = \l__fancythm_key_name_tl ,
126   name .default:n = \c_novalue_tl ,
127   mdframed .tl_set:N = \l__fancythm_key_mdframed_tl ,
128   mdframed .default:n = \c_novalue_tl ,
129   style .tl_set:N = \l__fancythm_key_style_tl ,
130   style .default:n = \c_novalue_tl ,
131   group .clist_set:N = \l__fancythm_key_group_clist ,
132   group .default:n = {} ,
133   thmtools .clist_set:N = \l__fancythm_key_thmtools_clist ,
134   thmtools .default:n = {} ,
135 }

```

Process the given keys:

```

136 \keys_set:nn { fancythm } { translator, generate defaults, number in, style }
137 \ProcessKeysOptions{ fancythm }

```

`__fancythm_require_package:nn`

```

138 \cs_new:Npn \__fancythm_require_package:nn #1
139 {
140   \RequirePackage [ #1 ]
141 }
142 \cs_generate_variant:Nn \__fancythm_require_package:nn { V n }

```

(End of definition for `__fancythm_require_package:nn`.)

```

143 \__fancythm_require_package:Vn \g__fancythm_groupthm_option_clist { groupthm }

```

This sets up translation if requested. Throughout implementation, we can just use `__fancythm_translate:n` and will (or not) have translation according to the specified options.

```

144 \bool_if:NTF \g__fancythm_translator_bool
145 {
146   \RequirePackage{translator}
147   \usedictionary{translator-environment-names}
148   \cs_set_eq:NN \__fancythm_translate:n \translate
149 }
150 {
151   \cs_set_eq:NN \__fancythm_translate:n \use:n
152 }

```

`__fancythm_set_normalized_keys:nn`

```

\__fancythm_set_normalized_keys:nn{<keys>}{<fallback name>}
153 \cs_new:Npn \__fancythm_set_normalized_keys:nn #1 #2

```

```

154 {
155   \keys_set:nn { fancythm / fancytheorem } { name, mdframed, style, group, thmtools }
156   \keys_set:nn { fancythm / fancytheorem } { #1 }
157   \clist_set_eq:NN \l__fancythm_group_clist \l__fancythm_key_group_clist
158   \clist_set_eq:NN \l__fancythm_thmtools_clist \l__fancythm_key_thmtools_clist
159   \tl_if_eq:NnF \l__fancythm_key_mdframed_tl { \c_novalue_tl }
160   {
161     \clist_put_right:Nx \l__fancythm_thmtools_clist
162     {
163       mdframed = { style = \tl_use:N \l__fancythm_key_mdframed_tl }
164     }
165   }
166   \tl_if_eq:NnF \l__fancythm_key_style_tl { \c_novalue_tl }
167   {
168     \clist_put_right:Nx \l__fancythm_thmtools_clist
169     {
170       style = \tl_use:N \l__fancythm_key_style_tl
171     }
172   }
173   \tl_if_eq:NnTF \l__fancythm_key_name_tl { \c_novalue_tl }
174   {
175     \tl_set:Nx \l__fancythm_name_tl
176     {
177       \text_titlecase_first:n { \tl_trim_spaces:n { #2 } }
178     }
179   }
180   {
181     \tl_set_eq:NN \l__fancythm_name_tl \l__fancythm_key_name_tl
182   }
183   \tl_set:Nx \l__fancythm_name_tl
184   {
185     \exp_not:N \csname __fancythm_translate:n \exp_not:N \endcsname { \tl_use:N \l__fancy
186   }
187 }

```

(End of definition for `__fancythm_set_normalized_keys:nn`.)

1.6 Fancy theorems

`__fancythm_wrap_multiple:nnn` `__fancythm_wrap_multiple:nnn{<declarator list>}{<function name>}{<code>}`
 Defines `<function name>`, which is assumed to contain `\declarator` by `<code>` for each declarator in `<declarator list>`.

```

188 \cs_new:Npn \__fancythm_wrap_multiple:nnn #1 #2 #3
189 {
190   \cs_set:Npn \__fancythm_map_aux:n ##1
191   {
192     \cs_new:cn { #2 }
193     {
194       #3
195     }
196   }
197   \clist_map_function:nN { #1 } \__fancythm_map_aux:n
198 }

```

(End of definition for `_fancythm_wrap_multiple:nnn`.)

```

\__fancythm_new_theorem:nnnn      \fancythm_new_theorem:nnnn{\fancy theorem}{\groups}
\__fancythm_provide_theorem:nnnn { \name } {\thmtools keys}
199 \__fancythm_wrap_multiple:nnn
200 { new, provide }
201 { fancythm_#1_theorem:nnnn }
202 {
203   \use:c { groupthm_#1_family:nnnnn } { ##1 } { dagger, star, nobraces } { ##3 } { ##4 } {
204   \use:c { groupthm_#1_family_options:nnnn }
205     { ##1 }
206     { !s !t+ !t- }
207     {
208       \IfBooleanT { ####1 }
209       {
210         \AddTheoremToGroup { star }
211       }
212       \IfBooleanT { ####2 }
213       {
214         \AddTheoremToGroup { dagger }
215       }
216       \IfBooleanT { ####3 }
217       {
218         \AddTheoremToGroup { nobraces }
219       }
220     }
221     { ##2 }
222   }
223 \cs_generate_variant:Nn \fancythm_new_theorem:nnnn { n V V V }
224 \cs_generate_variant:Nn \fancythm_provide_theorem:nnnn { n V V V }

```

(End of definition for `\fancythm_new_theorem:nnnn` and `\fancythm_provide_theorem:nnnn`. These functions are documented on page ??.)

```

\__fancythm_new_theorem:nn      \fancythm_new_theorem:nn{\key=value list}{\fancy theorem}
225 \__fancythm_wrap_multiple:nnn
226 { new, provide }
227 { fancythm_#1_theorem:nn }
228 {
229   \__fancythm_set_normalized_keys:nn { ##1 } { ##2 }
230   \use:c { fancythm_#1_theorem:nVVV }
231     { ##2 }
232     \l__fancythm_group_clist
233     \l__fancythm_name_tl
234     \l__fancythm_thmtools_clist
235   }

```

(End of definition for `\fancythm_new_theorem:nn`. This function is documented on page ??.)

`__fancythm_new_document_command:Nnn`
`__fancythm_new_document_command:cnn`

Private wrappers around `\NewDocumentCommand`.

```

236 \cs_new:Npn \__fancythm_new_document_command:Nnn #1 #2 #3
237 {
238   \NewDocumentCommand { #1 } { #2 } { #3 }
239 }
240 \cs_generate_variant:Nn \__fancythm_new_document_command:Nnn { c n n }

```

(End of definition for `_fancythm_new_document_command:Nnn`.)

```
\_fancythm_wrap_multiple_document:nnnn
    \_fancythm_wrap_multiple_document:nnnn{<declarator list>}{<function name>}{<arg
    spec>}{<code>}
```

This is very similar to `_fancythm_wrap_multiple:nnn`, except that it produces document commands. For this reason, `\declarator` and `\Declarator` are available to refer to the lower and upper-case versions of the current declarator.

```
241 \cs_new:Npn \_fancythm_wrap_multiple_document:nnnn #1 #2 #3 #4
242 {
243   \cs_set:Npn \_fancythm_map_aux:n ##1
244   {
245     \cs_set:Nn \_fancythm_Declarator: { \text_titlecase_first:n { ##1 } }
246     \_fancythm_new_document_command:cnn { #2 } { #3 } { #4 }
247   }
248   \clist_map_function:nN { #1 } \_fancythm_map_aux:n
249 }
```

(End of definition for `_fancythm_wrap_multiple_document:nnnn`.)

```
\NewFancyTheorem      \NewFancyTheorem{<key=value list>}{<fancy theorem>}
\ProvideFancyTheorem
250 \_fancythm_wrap_multiple_document:nnnn
251 { new, provide }
252 { \_fancythm_Declarator: FancyTheorem }
253 { 0{ } m }
254 {
255   \use:c { fancythm_#1_theorem:nn } { ##1 } { ##2 }
256 }
```

(End of definition for `\NewFancyTheorem` and `\ProvideFancyTheorem`. These functions are documented on page ??.)

1.7 Provided resources

We introduce various new 0theorem groups that help us to organize the document in a flexible way.

```
star
dagger
big
small
tiny
custom
257 \declaretheoremstyle[notebraces={}{\{}]{nobraces}
258 \groupthm_new_group:nnnnn { star } { } { * } { } { }
259 \groupthm_new_group:nnnnn { dagger } { } { } { $\dagger$ } { } { }
260 \groupthm_new_group:nnnnn { big } { } { } { } { } { sibling = toplevel }
261 \groupthm_new_group:nnnnn { small } { } { } { } { } { sibling = sublevel }
262 \groupthm_new_group:nnnnn { tiny } { } { } { } { } { numbered = no }
263 \groupthm_new_group:nnnnn { custom } { } { } { } { } { sibling = sublevel }
264 \groupthm_new_group:nnnnn { nobraces } { } { } { } { } { style = nobraces }
```

(End of definition for `star` and others. These variables are documented on page ??.)

```
265 \groupthm_add_parent:nn { star } { custom }
266 \groupthm_add_parent:nn { dagger } { custom }
```

```

267 \DeclareTheoremGroupRule [ suffix ] { dagger } { higher } { star }
268 \DeclareTheoremGroupRule { tiny } { higher } { small }
269 \DeclareTheoremGroupRule { tiny } { higher } { big }
270 \DeclareTheoremGroupRule { tiny } { higher } { custom }
271 \DeclareTheoremGroupRule { small } { higher } { big }
272 \DeclareTheoremGroupRule { custom } { higher } { big }

```

It remains to provide a list of theorems at the beginning of the document.

\fancythm_add_provided_theorem_to_group:nn

```

273 \cs_new:Npn \fancythm_add_provided_theorem_to_group:nn #1 #2
274 {
275   \cs_if_exist:cF { __fancythm_provided_theorem__#1__group_clist }
276   {
277     \clist_new:c { __fancythm_provided_theorem__#1__group_clist }
278   }
279   \clist_put_left:cn { __fancythm_provided_theorem__#1__group_clist } { #2 }
280 }

```

(End of definition for \fancythm_add_provided_theorem_to_group:nn. This function is documented on page ??.)

\AddProvidedFancyTheoremToGroup

```

281 \NewDocumentCommand { \AddProvidedFancyTheoremToGroup } { m m }
282 {
283   \fancythm_add_provided_theorem_to_group:nn { #1 } { #2 }
284 }

```

(End of definition for \AddProvidedFancyTheoremToGroup. This function is documented on page ??.)

__fancythm_provide_package_theorem:nnn

```

\__fancythm_provide_package_theorem:nnn{<key=value list>}{<fancy theorem>}
{<group>}

```

Group can be one of big, small, tiny, and the provided theorem will have this group, unless it conflicts with a user-provided group, in which case this is ignored.

```

285 \cs_new:Npn \__fancythm_provide_package_theorem:nnn #1 #2 #3
286 {
287   \__fancythm_set_normalized_keys:nn { #1 } { #2 }
288   \cs_if_exist:cT { __fancythm_provided_theorem__#2__group_clist }
289   {
290     \clist_concat:ccc
291     { l__fancythm_group_clist }
292     { __fancythm_provided_theorem__#2__group_clist }
293     { l__fancythm_group_clist }
294   }
295   \clist_if_in:NnF \l__fancythm_group_clist { big }
296   {
297     \clist_if_in:NnF \l__fancythm_group_clist { small }
298     {
299       \clist_if_in:NnF \l__fancythm_group_clist { tiny }
300       {
301         \clist_put_right:Nn \l__fancythm_group_clist { #3 }
302       }
303     }
304   }
305   \fancythm_provide_theorem:nVVV

```

```

306     { #2 }
307     \l__fancythm_group_clist
308     \l__fancythm_name_tl
309     \l__fancythm_thmtools_clist
310 }

```

(End of definition for __fancythm_provide_package_theorem:nnn.)

\fancythm_provide_big_theorem:nn

```

311 \cs_new:Npn \fancythm_provide_big_theorem:nn #1 #2
312 {
313     \__fancythm_provide_package_theorem:nnn { #1 } { #2 } { big }
314 }

```

(End of definition for \fancythm_provide_big_theorem:nn. This function is documented on page ??.)

\fancythm_provide_small_theorem:nn

```

315 \cs_new:Npn \fancythm_provide_small_theorem:nn #1 #2
316 {
317     \__fancythm_provide_package_theorem:nnn { #1 } { #2 } { small }
318 }

```

(End of definition for \fancythm_provide_small_theorem:nn. This function is documented on page ??.)

\fancythm_provide_tiny_theorem:nn

```

319 \cs_new:Npn \fancythm_provide_tiny_theorem:nn #1 #2
320 {
321     \__fancythm_provide_package_theorem:nnn { #1 } { #2 } { tiny }
322 }

```

(End of definition for \fancythm_provide_tiny_theorem:nn. This function is documented on page ??.)

```

323 \bool_if:NT \g__fancythm_generate_defaults_bool
324 {
325     \AddToHook { begindocument / before } [ fancythm ]
326     {
327         \fancythm_provide_big_theorem:nn { style = thmredmarginandfill } { theorem }
328         \fancythm_provide_big_theorem:nn { style = thmredmarginandfill } { proposition }
329         \fancythm_provide_big_theorem:nn { style = thmredmarginandfill } { corollary }
330         \fancythm_provide_big_theorem:nn { style = thmbluemarginandfill } { definition }
331         \fancythm_provide_big_theorem:nn { style = thmorangemarginandfill } { lemma }
332         \fancythm_provide_big_theorem:nn { style = thmgreenmargin } { example }
333         \fancythm_provide_big_theorem:nn
334         {
335             style = thmredmarginbluefill,
336             name = Theorem ~ and ~ Definition
337         } { theoremdef }
338         \fancythm_provide_big_theorem:nn
339         {
340             style = thmredmarginbluefill,
341             name = Proposition ~ and ~ Definition
342         } { propositiondef }
343         \fancythm_provide_big_theorem:nn
344         {
345             style = thmorangemarginbluefill,
346             name = Lemma ~ and ~ Definition
347         } { propositiondef }

```

```

348 \fancythm_provide_small_theorem:nn { style = thmvioletmargin } { notation }
349 \fancythm_provide_small_theorem:nn { style = thmyellowmargin } { remark }
350 \fancythm_provide_small_theorem:nn { style = thmgoldmargin } { praise }
351 \fancythm_provide_small_theorem:nn { style = thmblackmarginandfill } { question }
352 \fancythm_provide_small_theorem:nn { style = thmblackmargin } { orga }
353 \fancythm_provide_small_theorem:nn { style = thmredmargin } { fact }
354 \fancythm_provide_small_theorem:nn
355 {
356     style = thmyellowmargin,
357     name = Trivial ~ Nonsense
358 } { trivial }
359 \fancythm_provide_small_theorem:nn
360 {
361     style = thmvioletmargin,
362     name = Abuse ~ of ~ Notation
363 } { abuse }
364 \fancythm_provide_small_theorem:nn
365 {
366     style = thmyellowmargin,
367     name = Oral ~ remark
368 } { oral }
369 \fancythm_provide_tiny_theorem:nn { } { variant }
370 \fancythm_provide_tiny_theorem:nn { } { assumption }
371 \fancythm_provide_tiny_theorem:nn { } { note }
372 \fancythm_provide_tiny_theorem:nn { } { warning }
373 \fancythm_provide_tiny_theorem:nn { } { goal }
374 \fancythm_provide_tiny_theorem:nn { } { strategy }
375 \fancythm_provide_tiny_theorem:nn { } { problem }
376 \fancythm_provide_tiny_theorem:nn { } { info }
377 \fancythm_provide_tiny_theorem:nn { } { observe }
378 \fancythm_provide_tiny_theorem:nn { } { property }
379 \fancythm_provide_tiny_theorem:nn { } { intuition }
380 \fancythm_provide_tiny_theorem:nn { } { recall }
381 \fancythm_provide_tiny_theorem:nn { } { idea }
382 \fancythm_provide_tiny_theorem:nn { } { exercise }
383 \fancythm_provide_tiny_theorem:nn { } { reminder }
384 }
385 }
386 \DeclareHookRule { begindocument / before } { groupthm } { after } { fancythm }
387 \end{package}

```

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.

A	
<code>\AddProvidedFancyTheoremToGroup</code> . . .	<u>281</u>
<code>\AddTheoremToGroup</code>	<u>210</u> , <u>214</u> , <u>218</u>
<code>\AddToHook</code>	<u>325</u>
<code>\arabic</code>	<u>5</u>
arabic commands:	
<code>\arabic:n</code>	<u>5</u> , <u>5</u> , <u>51</u>
B	
<code>big</code>	<u>257</u>
bool commands:	
<code>\bool_if:NTF</code>	<u>144</u> , <u>323</u>
<code>\bool_new:N</code>	<u>83</u> , <u>84</u>
C	
clist commands:	
<code>\clist_concat:NNN</code>	<u>290</u>
<code>\clist_if_in:NnTF</code>	<u>295</u> , <u>297</u> , <u>299</u>
<code>\clist_map_function:nN</code>	<u>197</u> , <u>248</u>
<code>\clist_new:N</code>	<u>88</u> , <u>89</u> , <u>91</u> , <u>92</u> , <u>93</u> , <u>277</u>
<code>\clist_put_left:Nn</code>	<u>279</u>
<code>\clist_put_right:Nn</code>	<u>114</u> , <u>119</u> , <u>161</u> , <u>168</u> , <u>301</u>
<code>\clist_set_eq:NN</code>	<u>157</u> , <u>158</u>
counter commands:	
<code>\counter_alias:nn</code> <u>3</u> , <u>67</u> , <u>67</u> , <u>72</u> , <u>78</u> , <u>103</u>	
<code>\counter_ensure_exist:n</code>	<u>30</u> , <u>30</u> , <u>43</u>
<code>\counter_if_exist:NnTF</code>	<u>13</u> , <u>17</u> , <u>21</u> , <u>25</u> , <u>32</u> , <u>62</u>
<code>\counter_if_exist:TF</code>	<u>13</u>
<code>\counter_if_exist_p:n</code>	<u>13</u> , <u>13</u>
<code>\counter_new:n</code>	<u>5</u> , <u>6</u> , <u>10</u> , <u>56</u> , <u>73</u>
<code>\counter_new:nn</code>	<u>5</u> , <u>8</u>
<code>\counter_new_sub:nn</code> <u>54</u> , <u>64</u> , <u>79</u> , <u>80</u> , <u>81</u>	
<code>\counter_new_sub:nn#1_#2</code>	<u>54</u>
<code>\counter_set_the:nn</code>	<u>37</u> , <u>41</u> , <u>49</u>
<code>\counter_sub:nn</code>	<u>2</u> , <u>46</u> , <u>46</u> , <u>57</u>
<code>\counter_the:n</code>	<u>37</u> , <u>37</u> , <u>51</u>
<code>\counter_undefined_error:n</code> <u>29</u> , <u>29</u> , <u>34</u>	
<code>\counter_within:nn</code>	<u>5</u> , <u>7</u> , <u>11</u> , <u>48</u>
<code>\counterwithin</code>	<u>7</u>
cs commands:	
<code>\cs_generate_variant:Nn</code>	<u>72</u> , <u>142</u> , <u>223</u> , <u>224</u> , <u>240</u>
<code>\cs_if_exist:NnTF</code>	<u>19</u> , <u>23</u> , <u>27</u> , <u>275</u> , <u>288</u>
<code>\cs_if_exist_p:N</code>	<u>15</u>
<code>\cs_new:Nn</code>	<u>192</u>
<code>\cs_new:Npn</code>	<u>8</u> , <u>13</u> , <u>17</u> , <u>21</u> , <u>25</u> , <u>30</u> , <u>37</u> , <u>41</u> , <u>46</u> , <u>54</u> , <u>60</u> , <u>67</u> , <u>138</u> , <u>153</u> , <u>188</u> , <u>236</u> , <u>241</u> , <u>273</u> , <u>285</u> , <u>311</u> , <u>315</u> , <u>319</u>
<code>\cs_set:Nn</code>	<u>245</u>
<code>\cs_set:Npn</code>	<u>190</u> , <u>243</u>
<code>\cs_set_eq:NN</code>	<u>5</u> , <u>6</u> , <u>7</u> , <u>29</u> , <u>148</u> , <u>151</u>
<code>\cs_undefine:N</code>	<u>69</u>
<code>\csname</code>	<u>185</u>
<code>custom</code>	<u>257</u>
D	
<code>\dagger</code>	<u>259</u>
<code>dagger</code>	<u>257</u>
<code>\Declarator</code>	<u>8</u>
<code>\declarator</code>	<u>6</u> , <u>8</u>
<code>\DeclareHookRule</code>	<u>386</u>
<code>\DeclareTheoremGroupRule</code>	<u>267</u> , <u>268</u> , <u>269</u> , <u>270</u> , <u>271</u> , <u>272</u>
<code>\declaretheoremstyle</code>	<u>257</u>
E	
<code>\endcsname</code>	<u>185</u>
exp commands:	
<code>\exp_args:Nc</code>	<u>44</u>
<code>\exp_not:N</code>	<u>185</u>
F	
fancythm commands:	
<code>\fancythm_add_provided_theorem_-</code> <code>to_group:nn</code>	<u>273</u> , <u>273</u> , <u>283</u>
<code>\fancythm_new_theorem:nn</code>	<u>225</u>
<code>\fancythm_new_theorem:nnnn</code> <u>7</u> , <u>199</u> , <u>223</u>	
<code>\fancythm_provide_big_theorem:nn</code>	<u>311</u> , <u>311</u> , <u>327</u> , <u>328</u> , <u>329</u> , <u>330</u> , <u>331</u> , <u>332</u> , <u>333</u> , <u>338</u> , <u>343</u>
<code>\fancythm_provide_small_theorem:nn</code>	<u>315</u> , <u>315</u> , <u>348</u> , <u>349</u> , <u>350</u> , <u>351</u> , <u>352</u> , <u>353</u> , <u>354</u> , <u>359</u> , <u>364</u>
<code>\fancythm_provide_theorem:nnnn</code>	<u>199</u> , <u>224</u> , <u>305</u>
<code>\fancythm_provide_tiny_theorem:nn</code>	<u>319</u> , <u>319</u> , <u>369</u> , <u>370</u> , <u>371</u> , <u>372</u> , <u>373</u> , <u>374</u> , <u>375</u> , <u>376</u> , <u>377</u> , <u>378</u> , <u>379</u> , <u>380</u> , <u>381</u> , <u>382</u> , <u>383</u>
fancythm internal commands:	
<code>__fancythm_counter_sub_if_-</code> <code>exist:n</code>	<u>60</u> , <u>60</u> , <u>74</u> , <u>75</u> , <u>76</u> , <u>77</u>
<code>__fancythm_Declarator:</code>	<u>245</u> , <u>252</u>

<code>\g__fancythm_generate_defaults_-</code> <code>bool</code>	84, 98, 323		
<code>\l__fancythm_group_clist</code>	90, 157, 232, 295, 297, 299, 301, 307		
<code>\g__fancythm_groupthm_option_-</code> <code>clist</code>	93, 114, 119, 143		
<code>\l__fancythm_key_group_clist</code>	85, 131, 157		
<code>\l__fancythm_key_mdframed_tl</code>	85, 127, 159, 163		
<code>\l__fancythm_key_name_tl</code>	85, 125, 173, 181		
<code>\l__fancythm_key_style_tl</code>	85, 129, 166, 170		
<code>\l__fancythm_key_thmtools_clist</code>	85, 133, 158		
<code>__fancythm_map_aux:n</code>	190, 197, 243, 248		
<code>\l__fancythm_name_tl</code>	90, 175, 181, 183, 185, 233, 308		
<code>__fancythm_new_document_-</code> <code>command:Nnn</code>	236, 236, 240, 246		
<code>__fancythm_provide_package_-</code> <code>theorem:nnn</code>	9, 285, 285, 313, 317, 321		
<code>__fancythm_require_package:nn</code>	138, 138, 142, 143		
<code>__fancythm_set_normalized_-</code> <code>keys:nn</code>	5, 153, 153, 229, 287		
<code>\g__fancythm_style_str</code>	82, 109		
<code>\l__fancythm_thmtools_clist</code>	90, 158, 161, 168, 234, 309		
<code>__fancythm_translate:n</code>	5, 148, 151		
<code>\g__fancythm_translator_bool</code>	83, 96, 144		
<code>__fancythm_wrap_multiple:nnn</code>	6, 8, 188, 188, 199, 225		
<code>__fancythm_wrap_multiple_-</code> <code>document:nnnn</code>	8, 241, 241, 250		
fanythm commands:			
<code>\fanythm_new_theorem:nn</code>	7		
G			
groupthm commands:			
<code>\groupthm_add_parent:nn</code>	265, 266		
<code>\groupthm_new_group:nnnnn</code>	258, 259, 260, 261, 262, 263, 264		
I			
<code>\IfBooleanT</code>	208, 212, 216		
<code>inchapter</code>	73		
<code>indocument</code>	73		
<code>inpart</code>	73		
<code>insection</code>	73		
<code>insubsection</code>	73		
K			
keys commands:			
<code>\l_keys_choice_tl</code>	103, 109		
<code>\keys_define:nn</code>	94, 123		
<code>\keys_set:nn</code>	136, 155, 156		
N			
<code>\newcounter</code>	6		
<code>\NewDocumentCommand</code>	7, 238, 281		
<code>\NewFancyTheorem</code>	8, 250		
P			
<code>\ProcessKeysOptions</code>	137		
<code>\ProvideFancyTheorem</code>	250		
R			
<code>\renewcommand</code>	44		
<code>\RequirePackage</code>	3, 4, 140, 146		
S			
<code>small</code>	257		
<code>star</code>	257		
str commands:			
<code>\str_new:N</code>	82		
<code>\str_set:Nn</code>	109		
<code>sublevel</code>	78		
<code>subsublevel</code>	78		
<code>subsubsublevel</code>	78		
T			
TeX and L ^A T _E X 2 _ε commands:			
<code>\@counteralias</code>	3		
<code>\c@⟨countername⟩</code>	3		
text commands:			
<code>\text_titlecase_first:n</code>	177, 245		
<code>tiny</code>	257		
tl commands:			
<code>\c_novalue_tl</code>	126, 128, 130, 159, 166, 173		
<code>\tl_if_eq:NnTF</code>	159, 166, 173		
<code>\tl_new:N</code>	85, 86, 87, 90		
<code>\tl_set:Nn</code>	175, 183		
<code>\tl_set_eq:NN</code>	181		
<code>\tl_trim_spaces:n</code>	177		
<code>\tl_use:N</code>	103, 109, 163, 170, 185		
<code>toplevel</code>	78		
<code>\translate</code>	148		
U			
use commands:			
<code>\use:N</code>	39, 70, 203, 204, 230, 255		
<code>\use:n</code>	151		
<code>\usedictionary</code>	147		