

oops, an object oriented practical scribe's package.*

Erwann Rogard†

Released 2020/03/11

Abstract

`oops` is a package for L^AT_EX (hence “scribe”) that organizes (typically mathematical) definitions along two dimensions: functions and objects, hence “OO”. Such definitions are made *inline* thanks to a minimalist interface built upon `xparse`[3]. To make a definition, use `\OopsNew{t11}`, where `t11` identifies an object, followed by input that alternates between ‘text’ and instructions. The latter create and expand definitions using rules that can be modified at the package level or themselves inline. This framework is suitable for instance where `t11` is either of `ModelA` and `ModelB`, and each requires its own definition of, say, a *space*. In this case, they would be encoded respectively as `\Space{ModelA}` and `\Space{ModelB}`. However, this would be verbose if most functions applied to just one object, so the package provides a generic one that is set by default to `Math`. For example, `\OopsNew[Math]{Let-}{Space=\Omega}{~denote the sample space}`, followed by `\Space`, expand to: “Let Ω denote the sample space” and “ Ω ”. Other features automate repetitive formatting tasks. Altogether, “practical”.

Contents

| | | |
|----------|--------------------------------|----------|
| I | Usage | 3 |
| 1 | Convention | 3 |
| 2 | Loading the package | 4 |
| 3 | \OopsOptions | 4 |
| 3.1 | GenericObject | 4 |
| 3.2 | Inner | 4 |
| 3.3 | Separators | 4 |
| 3.4 | Outer | 4 |
| 4 | \OopsClear | 4 |

*This file describes version v1.0, last revised 2020/03/11.

†firstname dot lastname AusTria gmail dot com

| | | |
|-------------------------|---|-----------|
| 5 | \OoopsNew | 5 |
| 5.1 | { $\langle t11 \rangle$ } | 5 |
| 5.2 | [$\langle t12 \rangle$] | 5 |
| 5.3 | i{ $\langle code1 \rangle$ } | 5 |
| 5.4 | s{{ $\langle t13 \rangle$ }{{ $\langle t14 \rangle$ }{{ $\langle t15 \rangle$ }}} | 5 |
| 5.5 | o{ $\langle code2 \rangle$ } | 5 |
| 5.6 | { $\langle kv11 \rangle$ } | 5 |
| 5.7 | i{ $\langle code3 \rangle$ } | 6 |
| 5.8 | { $\langle kv12 \rangle$ } | 6 |
| 5.9 | [$\langle t16 \rangle$] | 6 |
| 6 | \langle Key | 6 |
| | | |
| II | Listings | 7 |
| Listing 1. | | 7 |
| Listing 2. | | 7 |
| Listing 3. | | 7 |
| Listing 4. | | 7 |
| Listing 5. | | 7 |
| Listing 6. | | 8 |
| Listing 7. | | 8 |
| Listing 8. | | 8 |
| | | |
| III | Other | 9 |
| 1 Acknowledgment | | 9 |
| 2 Bug | | 10 |
| 3 Disclaimer | | 10 |
| 4 Support | | 10 |
| 5 To do | | 10 |
| | | |
| IV | Implementation | 11 |

| | | |
|-----------------------|-------------------------|-----------|
| 1 | Back end | 11 |
| 1.1 | Aux | 11 |
| 1.1.1 | Msg | 11 |
| 1.1.2 | Variables | 11 |
| 1.1.3 | Options | 11 |
| 1.2 | Prop | 11 |
| 1.2.1 | name | 11 |
| 1.2.2 | new | 11 |
| 1.2.3 | put | 12 |
| 1.2.4 | item | 12 |
| 1.2.5 | conditional | 12 |
| 1.2.6 | parse | 13 |
| 2 | Front end | 13 |
| 2.1 | \OoopsOptions | 13 |
| 2.2 | \OoopsClear | 13 |
| 2.3 | \OoopsNew | 13 |
| Change History | | 15 |
| Index | | 15 |

Part I

Usage

Convention

- a) By default, all commands are declared in the *body* of \documentclass.
- b) Arguments expecting a *token list*[5], *keyval list*[2, l3keys], a character, and *inline code*, are denoted respectively $\langle tl \rangle$, $\langle kvl \rangle$, $\langle char \rangle$, and $\langle code \rangle$.
- c) $\{\langle arg \rangle\}$ is either m or g options[3], and $[\langle arg \rangle]$ and $\langle char \rangle\{\langle arg \rangle\}$ are o and e options[3].
- d) If we say that $\langle option \rangle$ can be used to override $\langle default \rangle$, and $\langle option \rangle$ is no value[3], we will treat it as though $\langle option \rangle = \langle default \rangle$.
- e) We say “set the key to” as shorthand for “set the value associated with the key to”
- f) We call \code{\langle arg \rangle}, $\langle code \rangle$ with #1 replaced by $\{\langle arg \rangle\}$

The template for the description of functions and arguments is, where applicable:

Use it to

Requirement

Side effect

Expands to

Default

Example

Other

Only the items that cannot be deduced from other information, are given. For instance, the requirement that $\langle t11 \rangle$ be a *token list* is omitted.

```
\usepackage \usepackage[kv10]{oops}
```

Use it to Load the package

Requirement 1. `oops.sty` is in the path of the L^AT_EX engine. See [Part III, section 4](#).
2. Declared in the *preamble* of `\documentclass`

Side effect That of \OoopsOptions{ $\langle kv10 \rangle$ }

`\OoopsOptions{<kv10>}`

Use it to Set default options for \OoopsNew

Other Also works in the *preamble*

$\langle kvl \rangle$

Requirement Keys listed below.

GenericObject

Requirement See \OopsNew, $\langle t11 \rangle$

Default Math

Inner

Requirement See \0oopsNew, ⟨code1⟩

Default {#1}

Separators

Requirement See [2, Section 8 of l3seq]

Default {\text{~and~}}{\text{,~}}{\text{~and~}}

Outer

Requirement See \0opsNew, ⟨code2⟩

Default \ensuremath{\{#1\}}

\OoopsClear \OoopsClear{\langle t11\rangle}

Side effect Clears $\langle t11 \rangle$ of any $\langle data \rangle$

\OopsNew $\backslash\text{OopsNew}\{\langle t11 \rangle\}$
 $\langle t12 \rangle$
 $i\{\langle code1 \rangle\}$
 $s\{\{\langle t13 \rangle\}\{\langle t14 \rangle\}\{\langle t15 \rangle\}\}$
 $o\{\langle code2 \rangle\}$
 $\{\langle kv11 \rangle\}$
 $i\{\langle code3 \rangle\}$
 $\{\langle kv12 \rangle\}$
 $[\langle t16 \rangle]$

Requirement Only $\langle t11 \rangle$ and $\langle kv11 \rangle$ are mandatory

$\langle t11 \rangle$

Use it to Identify an object

Side effect Registers $\langle t11 \rangle$ as a new object, if applicable

Example Math, ModelA, ModelB

$\langle t12 \rangle$

Use it to Bring about a definition

Expands to $\langle t12 \rangle$

Example Let~

$\langle code1 \rangle$

Use it to Override Inner.

Example $\backslash\mathbb{\#1}$

$\{\langle t13 \rangle\}\{\langle t14 \rangle\}\{\langle t15 \rangle\}$

Use it to Override Separators

Example $\{\sim\&\sim\}\{\sim\}\{\sim\&\sim\}$

$\langle code2 \rangle$

Use it to Override Outer

Example $\backslash\text{text}\{\#1\}$

$\langle kv11 \rangle$

Side effect If Key is a new key, attaches to it $\langle data \rangle = \langle code1 \rangle\{Value\}$

Expands to If $\langle t12 \rangle$ is no value, none, otherwise,

1. For each Key, calls $\backslash\text{Key}\langle t11 \rangle$. Call it $\langle seq \rangle$.

2. Concatenates $\langle seq \rangle$ using $\{\langle t13 \rangle\}\{\langle t14 \rangle\}\{\langle t15 \rangle\}$. Call it $\langle tmp \rangle$.
3. Expands $\backslash\langle code2 \rangle\{\langle tmp \rangle\}$

Example `Sample=\Omega`

The remaining options taken together forward to:
 $\backslash\text{O}\text{o}\text{ps}\text{N}\text{ew}\{\langle t11 \rangle\} \text{ o}\{\langle code3 \rangle\} \{\langle kv12 \rangle\} [\langle t16 \rangle]$

$\langle code3 \rangle$
 $\langle kv12 \rangle$
 $\langle t16 \rangle$

$\backslash<\text{Key}>$ $\backslash\langle Key \rangle[\langle t11 \rangle]$

Side effect Expands to $\langle data \rangle$ associated with $\langle t11 \rangle$ and $\langle Key \rangle$, if applicable.

Part II

Listings

Listing 1.

```
%      \OoopsOptions
%
%      {
%      Inner = {#1},
%      Outer = {#1}
%
%
```

Listing 2.

```
%      \OoopsNew{Foo}{ Barr = { a }, Baz = { b } }{ Qux = { c } }
%      \textless(\Barr[ Foo ]), (\Baz[ Foo ]) \& (\Qux[ Foo ]) \textgreater
%
<( a ), ( b ) & ( c )>
```

Listing 3.

```
%      \OoopsNew{Foo}[] if(#1) s{{~\&~}{~,~}{~\&~}} o{\textless\#1\textgreater}{%
%      Barr = { a }, Baz = { b }, Qux = { c } }
%
<( a ), ( b ) & ( c )>
```

Listing 4.

```
%      \OoopsOptions
%
%      {
%      Inner = {(#1)},
%      Separators = { {~\&~}{~,~}{~\&~} },
%      Outer = {\textless\#1\textgreater}
%
%      \OoopsNew{Foo}[]{ Barr = { a }, Baz = { b }, Qux = { c } }
%
```

<(a), (b) & (c) >

Listing 5.

```
%      \OoopsOptions
%
```

```

%      Inner,
%      Separators,
%      Outer
%
}
%
```

Listing 6.

```

%      \OoopsNew{Math}[We call~]{Elems=\{\omega_1, \dots, \omega_n\}}
%      [~the elementary events, and ]{}{Space=\Omega}
%      [\begin{equation*}\Space=(\Elems)\end{equation*}~the sample space.]
%
%
```

We call $\omega_1, \dots, \omega_n$ the elementary events, and

$$\Omega = (\omega_1, \dots, \omega_n)$$

the sample space.

Listing 7.

```

%      \OoopsClear{Math}
%      \OoopsNew{Math}[Let ]s{{},{},{},}{o{\ensuremath{\{\#1\}}}}
%      {Space=\Omega, SigmaField=\mathcal{F}, Measure=\mathcal{P}}
%      [~denote the probability space, where $\SigmaField\subset
2^{\{\Space\}.}]
%
{}
```

Let $\{\Omega, \mathcal{F}, \mathcal{P}\}$ denote the probability space, where $\mathcal{F} \subset 2^\Omega$.

Listing 8.

```

%      \newtheorem{theorem}{Theorem}
%      \OoopsNew{Math}
%      i{\mathbb{#1}}{ N = \{ N \} , R = \{ R \} }
%      [\begin{theorem}[Mittelwertsatz f\"ur $n$ Variable]Es sei~]
%      {}
%      {
%          OffeneMenge={D},
%          Ci={C^{#1}},
%          Strecke=\{[x_0,x]\}
%          %Strecke=\{\char`[x_0,x]\} % Also works
%          %Strecke=\{[x_0,x]\} % BUG
%      }
%      [$n\in N$, $\OffeneMenge\subseteq N$ eine offene Menge und $f\in
\mathcal{C}(\OffeneMenge, R)$.
```

```

% Dann gibt es auf jeder Strecke $\Strecke \subset \OffeneMenge$ einen
% Punkt $\xi \in \Strecke$, ]
%
%
% yDifferenz={f(x)-f(x_0)},
% xDifferenz={x-x_0},
% Steigung={\frac{yDifferenz}{xDifferenz}}
%
% [so dass gilt \begin{equation*}Steigung = \operatorname{grad} f(\xi)^\top
% \end{equation*}
% \end{theorem}]
%
%
```

Theorem 1 (Mittelwertsatz für n Variable) Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{R}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt

$$\frac{f(x) - f(x_0)}{x - x_0} = \operatorname{grad} f(\xi)^\top$$

Part III Other

1 Acknowledgment

This work has benefited from Q&A's from the L^AT_EX community. For specifics, see here: <https://tex.stackexchange.com/users/112708/erwann?tab=questions> Listing 6 and Listing 7 are from [1]. Listing 8 is from tcolorbox[4, 17.3].

References

- [1] A.N. Shiryaev *Probability* Springer, 1995
- [2] The L^AT_EX3 Project Team *The L^AT_EX3 interfaces* <http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf>
- [3] The L^AT_EX3 Project Team *The xparse package* <http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3packages/xparse.pdf>
- [4] Thomas F. Sturm *The tcolorbox package* <http://www.texdoc.net/texmf-dist/doc/latex/tcolorbox/tcolorbox.pdf>
- [5] <https://tex.stackexchange.com/questions/104023/what-is-a-token#104025>

2 Bug

See Listing 7. Low priority as there are workarounds.

3 Disclaimer

This package has not been tested beyond Part II.

Option type `G` is supported but not recommended by `xparse[3]`, but it's really practical in this case.

4 Support

This package is available from <https://www.ctan.org/pkg/oops> (release) or <https://github.com/rogard/oops> (development) where you can report issues.

5 To do

1. Continue Part II based on [4, Section 17.3]

Part IV

Implementation

1 Back end

1.1 Aux

1.1.1 Msg

```
1 \NeedsTeXFormat{LaTeX2e}[2019/10/01]
2 \ExplSyntaxOn
3 \msg_new:nnn
4 {Ooops}
5 {generic}
6 {#1}
```

1.1.2 Variables

```
7 \seq_new:N \__ erw_oops_seq
```

1.1.3 Options

```
8 \keys_define:nn { Ooops }
9 {
10   GenericObject .tl_gset:N = \__ erw_oops_object_default_tl,
11   GenericObject .value_required:n = false,
12   GenericObject .default:n = {Math},
13   GenericObject .initial:n = {Math},
14   Inner .code:n =
15   {
16     \cs_gset:Npn \__ erw_oops_inner_default:n ##1{#1}
17   },
18   Inner .value_required:n = false,
19   Inner .default:n = {#1},
20   Inner .initial:n = {#1},
21   Separators .tl_gset:N = \__ erw_oops_separators_default_tl,
22   Separators .value_required:n = false,
23   Separators .default:n = {{ \text{~and~}}{\text{,~}}{\text{,~and~}}},
24   Separators .initial:n = {{ \text{~and~}}{\text{,~}}{\text{,~and~}}},
25   Outer .code:n =
26   {
27     \cs_gset:Npn \__ erw_oops_outer_default:n ##1{#1}
28   },
29   Outer .value_required:n = false,
30   Outer .default:n = {\ensuremath{#1}},
31   Outer .initial:n = {\ensuremath{#1}}
32 }
```

1.2 Prop

1.2.1 name

```
33 \cs_new:Npn \__ erw_oops_name:n #1{__ erw_oops_#1}
```

1.2.2 new

```

34 \cs_new_protected:Nn \__ erw_oops_new:n
35 {
36   \prop_new:c{\__ erw_oops_name:n { #1 }}
37 }
38 \cs_new_protected:Nn \__ erw_oops_clear_new:n
39 {
40   \prop_clear_new:c{\__ erw_oops_name:n { #1 }}
41 }

```

1.2.3 put

```

42 \cs_new_protected:Nn \__ erw_oops_put:nnn
43 {
44   \prop_put:cnn { \__ erw_oops_name:n { #1 } } { #2 } { #3 }
45 }
46 \cs_new_protected:Nn \__ erw_oops_putinner:nnn
47 {
48   \__ erw_oops_put:nnn{#1}
49   {#2}
50   {\__ erw_oops_inner:n{#3}}
51 }

```

1.2.4 item

```

52 \cs_new:Nn \__ erw_oops_item:nn
53 {
54   \prop_item:cn { \__ erw_oops_name:n { #1 } } { #2 }
55 }
56 \cs_new:Npn \__ erw_oops_item:nnn #1 #2 #3
57 {
58   \__ erw_oops_item:nn{#1}{#2}
59 }
60 \cs_new:Nn \__ erw_oops_itemcmd:Nn
61 {
62   \ProvideDocumentCommand{#1}
63   {O{\__ erw_oops_object_default_tl}}
64   {
65     \__ erw_oops_item:nn {##1}{ #2 }
66   }
67 }
68 \cs_generate_variant:Nn \__ erw_oops_itemcmd:Nn {c}
69 \cs_new:Npn \__ erw_oops_itemcmd:nnn
70 #1
71 #2
72 #3
73 {
74   \__ erw_oops_itemcmd:cn{#2}{#2}
75 }
76 \cs_new:Npn \__ erw_oops_itemto_seq:nnn #1 #2 #3
77 {
78   \seq_put_right:Nn
79   \__ erw_oops_seq
80   {\__ erw_oops_item:nn{#1}{#2}}
81 }

```

1.2.5 conditional

```

82 \cs_new:Nn \__erw_oops_if_exist:nTF
83 {
84   \prop_if_exist:cTF{\__erw_oops_name:n { #1 }}{#2}{#3}
85 }

1.2.6 parse

86 \cs_set:Npn \__erw_oops_parse:Nnn
87 #1 % fun
88 #2 % prop name
89 #3 % clist
90 {
91
92   \tl_if_blank:nTF{#3}
93   {\c_empty_tl}
94   {
95     \seq_set_split:Nnn \l_tmpa_seq {,}{#3}
96     \cs_set:Npn \__erw_oops_parse:w ##1 = ##2 \q_stop
97     {
98       #1
99       {#2}
100      {\tl_trim_spaces:n{##1}} % key
101      {\tl_trim_spaces:n{##2}} % value
102    }
103    \cs_set:Npn \__erw_oops_parse:n ##1
104    {
105      \__erw_oops_parse:w ##1 \q_stop
106    }
107    \seq_map_function:NN \l_tmpa_seq \__erw_oops_parse:n
108  }
109 }

```

2 Front end

2.1 \OopsOptions

```

110 \NewDocumentCommand{\OopsOptions}
111 {m}
112 {
113   \keys_set:nn { Oops } {#1}
114 }
115 % \ProcessKeysPackageOptions{ Oops }

```

2.2 \OopsClear

```

116 \NewDocumentCommand{\OopsClear}
117 {m}
118 {
119   \__erw_oops_clear_new:n{#1}
120 }

```

2.3 \OopsNew

```

121 \NewDocumentCommand{\OopsNew}
122 {
123   m
124   +o

```

```

125 E{iso}
126 {
127     {\_\_ erw\_oops\_inner\_default:n{\#\#1}}
128     {\_\_ erw\_oops\_separators\_default\_tl}
129     {\_\_ erw\_oops\_outer\_default:n{\#\#1}}
130 }
131 m
132 E{i}
133 {
134     {\_\_ erw\_oops\_inner\_default:n{\#\#1}}
135 }
136 g
137 +o
138 }
139 {
140     \_\_ erw\_oops\_if\_exist:nTF{\#1}
141     {\c_empty_tl}
142     {\_\_ erw\_oops\_new:n{\#1}}
143
144     \IfValueTF{\#2}
145     {\#2}
146     {\c_empty_tl}
147     \cs_gset:Npn
148
149     \_\_ erw\_oops\_inner:n ##1
150     {\#3}
151     \_\_ erw\_oops\_parse:Nnn
152     \_\_ erw\_oops\_putinner:nnn
153     {\#1}
154     {\#6}
155
156     \_\_ erw\_oops\_parse:Nnn
157     \_\_ erw\_oops\_itemcmd:nnn
158     {\#1}
159     {\#6}
160
161     \IfValueTF{\#2}
162     {
163         \seq_clear:N \_\_ erw\_oops\_seq
164         \_\_ erw\_oops\_parse:Nnn
165         \_\_ erw\_oops\_itemto\_seq:nnn
166         {\#1}
167         {\#6}
168         \cs_gset:Npn
169         \_\_ erw\_oops\_outer:n ##1
170         {\#5}
171         \_\_ erw\_oops\_outer:n
172         {\exp_last_unbraced:NNo
173             \seq_use:Nnnn
174             \_\_ erw\_oops\_seq
175             {\#4}
176         }
177     }
178     {\c_empty_tl}

```

```

179   \IfValueTF{#8}
180   {
181     \IfValueTF{#9}
182     {
183       \exp_not:n{\OoopsNew{#1}i{#7}{#8}[#9]}
184     }
185     {
186       \exp_not:n{\OoopsNew{#1}i{#7}{#8}}
187     }
188   }
189   {
190     \IfValueTF{#9}
191     {
192       \exp_not:n{\OoopsNew{#1}[#9]}
193     }
194     {\c_empty_tl}
195   }
196 }
197 }
198 \ExplSyntaxOff

```

Change History

v1.0
 General: Initial version 10

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 | | |
|-----------------|--------------|---|
| \<Key> | 6 | |
| \<Key> | <u>2</u> , 6 | C |
| \<kv1> (option) | 4 | |
| \<code1> | 5 | \code 3 |
| \<code2> | 6 | cs commands: |
| \Key | 5 | \cs_generate_variant:Nn 68 |
| \code1 (option) | 5 | \cs_gset:Npn 16, 27, 147, 168 |
| \code2 (option) | 5 | \cs_new:Nn 52, 60, 82 |
| \code3 (option) | 6 | \cs_new:Npn 33, 56, 69, 76 |
| \kv11 (option) | 5 | \cs_new_protected:Nn . . . 34, 38, 42, 46 |
| \kv12 (option) | 6 | \cs_set:Npn 86, 96, 103 |
| \t11 (option) | 5 | |
| \t12 (option) | 5 | D |
| \t16 (option) | 6 | \documentclass 3, 4 |

| | | |
|----------------------------------|-------------------------|--|
| | E | |
| \ensuremath | 30, 31 | |
| erw internal commands: | | |
| __erw_oops_clear_new:n | 38, 119 | |
| __erw_oops_if_exist:nTF | 82, 140 | |
| __erw_oops_inner:n | 50, 149 | |
| __erw_oops_inner_default:n | 16, 127, 134 | |
| __erw_oops_item:nn | 52, 58, 65, 80 | |
| __erw_oops_item:nnn | 56 | |
| __erw_oops_itemcmd:Nn | 60, 68, 74 | |
| __erw_oops_itemcmd:nnn | 69, 157 | |
| __erw_oops_itemto_seq:nnn | 76, 165 | |
| __erw_oops_name:n | 33, 36, 40, 44, 54, 84 | |
| __erw_oops_new:n | 34, 142 | |
| __erw_oops_object_default_tl | 10, 63 | |
| __erw_oops_outer:n | 169, 171 | |
| __erw_oops_outer_default:n | 27, 129 | |
| __erw_oops_parse:n | 103, 107 | |
| __erw_oops_parse:Nnn | 86, 151, 156, 164 | |
| __erw_oops_parse:w | 96, 105 | |
| __erw_oops_put:nnn | 42, 48 | |
| __erw_oops_putinner:nnn | 46, 152 | |
| __erw_oops_separators_default_- | | |
| tl | 21, 128 | |
| __erw_oops_seq | 7, 79, 163, 174 | |
| exp commands: | | |
| \exp_last_unbraced:NNo | 172 | |
| \exp_not:n | 184, 187, 193 | |
| \ExplSyntaxOff | 198 | |
| \ExplSyntaxOn | 2 | |
| | G | |
| GenericObject (option) | 4 | |
| | I | |
| \IfValueTF | 144, 161, 180, 182, 191 | |
| Inner (option) | 4 | |
| | K | |
| keys commands: | | |
| \keys_define:nn | 8 | |
| \keys_set:nn | 113 | |
| | M | |
| msg commands: | | |
| \msg_new:nnn | 3 | |
| | N | |
| \NeedsTeXFormat | 1 | |
| \NewDocumentCommand | 110, 116, 121 | |
| | O | |
| \OoopsClear | 1, 3, 4, 13, 116 | |
| | P | |
| \ProcessKeysPackageOptions | 115 | |
| prop commands: | | |
| \prop_clear_new:N | 40 | |
| \prop_if_exist:NTF | 84 | |
| \prop_item:Nn | 54 | |
| \prop_new:N | 36 | |
| \prop_put:Nnn | 44 | |
| \ProvideDocumentCommand | 62 | |
| | Q | |
| quark commands: | | |
| \q_stop | 96, 105 | |
| | S | |
| Separators (option) | 4 | |
| seq commands: | | |
| \seq_clear:N | 163 | |
| \seq_map_function:NN | 107 | |
| \seq_new:N | 7 | |
| \seq_put_right:Nn | 78 | |
| \seq_set_split:Nnn | 95 | |
| \seq_use:Nnnn | 173 | |
| \l_tmpa_seq | 95, 107 | |
| \Space | 1 | |
| | T | |
| \text | 23, 24 | |
| tl commands: | | |
| \c_empty_tl | 93, 141, 146, 178, 195 | |
| \tl_if_blank:nTF | 92 | |
| \tl_trim_spaces:n | 100, 101 | |
| | U | |
| \usepackage | 4 | |