Abstract

\texttt{oops} is a package for \LaTeX (hence “scribe”) that organizes (typically mathematical) definitions along two dimensions: functions and objects, hence “OO”. Such definitions are made \textit{inline} thanks to a minimalist interface built upon \texttt{xparse}[3].

To make a definition, use \texttt{\OopsNew{(t1)}}}, where \texttt{(t1)} 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 \texttt{(t1)} is either of \texttt{ModelA} and \texttt{ModelB}, and each requires its own definition of, say, a \textit{space}. In this case, they would be encoded respectively as \texttt{\Space{ModelA}} and \texttt{\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 \texttt{Math}. For example, \texttt{\OopsNew{Math}[Let\text{-}][Space=\Omega][\text{-}denote the sample space]{}}, followed by \texttt{$\Space$}, expand to: “Let $\Omega$ denote the sample space” and “$\Omega$”. Other features automate repetitive formatting tasks. Altogether, “practical”.

Contents

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

---

*This file describes version v1.0, last revised 2020/03/11.
†firstname dot lastname AusTria gmail dot com
II Listings
Listing 1.
Listing 2.
Listing 3.
Listing 4.
Listing 5.
Listing 6.
Listing 7.
Listing 8.

III Other
1 Acknowledgment
2 Bug
3 Disclaimer
4 Support
5 To do

IV Implementation
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 \texttt{⟨tl⟩}, ⟨kvl⟩, ⟨char⟩, and ⟨code⟩.

c) \texttt{⟨arg⟩} is either \texttt{m} or \texttt{g} options[^3], and \texttt{⟨char⟩} and \texttt{(char)⟨arg⟩} are \texttt{o} and \texttt{e} options[^3].

d) If we say that \texttt{(option)} can be used to override \texttt{(default)}, and \texttt{(option)} is no value[^3], we will treat it as though \texttt{(option)=⟨default⟩}.

e) We say “set the key to” as shorthand for “set the value associated with the key to”

f) We call \texttt{\textbackslash code⟨arg⟩}, ⟨code⟩ with \texttt{#1} replaced by \texttt{⟨arg⟩}.

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

Use it to

Requirement

Side effect

[^5]: token list
[^2]: keyval list
[^3]: options
Expands to
Default
Example
Other

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

\usepackage[⟨kvl0⟩]{oops}

Use it to Load the package

Requirement 1. oops.sty is in the path of the LaTeX engine. See Part III, section 4.
2. Declared in the preamble of \documentclass

Side effect That of \OopsOptions{⟨kvl0⟩}

\OopsOptions{⟨kvl0⟩}

Use it to Set default options for \OopsNew

Other Also works in the preamble

⟨kvl⟩

Requirement Keys listed below.

GenericObject

Requirement See \OopsNew, ⟨t11⟩

Default Math

Inner

Requirement See \OopsNew, ⟨code1⟩

Default {#1}

Separators

Requirement See [2, Section 8 of l3seq]

Default \{\text{-and-}\{\text{-}\}{\text{-and-}}\}

Outer

Requirement See \OopsNew, ⟨code2⟩

Default \ensuremath{#1}

\OopsClear{⟨t11⟩}
**Side effect** Clears \(<t11>\) of any \(<data>\)

\(\text{\texttt{\textbackslash OopsNew(t11)}}\)[[t12]][{tl1}]
  \texttt{if(\texttt{\textbackslash code1})}
  \texttt{s\{\{tl3\}\{\{tl4\}\{\{tl5\}\}\}}
  \texttt{of(\texttt{\textbackslash code2})}
  \texttt{(\texttt{kv11})}
  \texttt{if(\texttt{\textbackslash code3})}
  \texttt{(\texttt{kv12})}
  \texttt{[\{t16\}]}

**Requirement** Only \(<t11>\) and \(<kv11>\) are mandatory

\(<t11>\)

**Use it to** Identify an object

**Side effect** Registers \(<t11>\) as a new object, if applicable

**Example** Math, ModelA, ModelB

\(<t12>\)

**Use it to** Bring about a definition

**Expands to** \(<t12>\)

**Example** Let~

\(<\texttt{\textbackslash code1}>\)

**Use it to** Override Inner.

**Example** \texttt{\textbackslash mathbb\{#1\}}

\(<\{tl3\}\{\{tl4\}\{\{tl5\}\}\}>\)

**Use it to** Override Separators

**Example** \{-\&-\}{,-}{-\&-}

\(<\texttt{\textbackslash code2}>\)

**Use it to** Override Outer

**Example** \texttt{\textbackslash text\{#1\}}

\(<kv11>\)

**Side effect** If Key is a new key, attaches to it \(<data>=\texttt{\textbackslash code1}\{Value}\>

**Expands to** If \(<t12>\) is no value, none, otherwise.

1. For each Key, calls \texttt{\textbackslash Key\{t11\}}. Call it \(<seq>\).
2. Concatenates \( \langle seq \rangle \) using \{\langle tl3 \rangle \}{\langle tl4 \rangle \}{\langle tl5 \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:
\( \backslash \text{OopsNew}\{\langle tl1 \rangle \} \ o\{\langle code3 \rangle \} \{\langle kv12 \rangle \} \{\langle tl6 \rangle \} \)

\( \langle code3 \rangle \langle kvl2 \rangle \langle tl6 \rangle \)

\( \langle Key \rangle \langle tl1 \rangle \)

\( \langle Key \rangle \{\langle tl1 \rangle \} \)

**Side effect** Expands to \( \langle data \rangle \) associated with \( \langle tl1 \rangle \) and \( \langle Key \rangle \), if applicable.
Part II
Listings

Listing 1.
\% \OopsOptions
\%
\{% Inner = {#1},
\%
\Outer = {#1}
\%
\}
\%

Listing 2.
\% \OopsNew{Foo}{\textless (\Barr[foo]), (\Baz[foo]) \& (\Qux[foo])\textgreater
\%
%( a ), ( b ) \& ( c )>\n
Listing 3.
\% \OopsNew{Foo}\{\textless (#1)\textgreater\{\textless-\&-\textgreater\{\textless-\&-\textgreater\}o\textless-\&-\textgreater\}o{\textless#1\textgreater\}
\%
%( a ), ( b ) \& ( c )>\n
Listing 4.
\% \OopsOptions
\%
\{% Inner = {(#1)},
\%
\Separators = {\textless-\&-\textgreater\{-\&-\textgreater\}},
\%
\Outer = {\textless#1\textgreater\}
\%
\}
\% \OopsNew{Foo}\{\textless (\Barr, Baz = ( b )}, Qux = ( c )\}
\%
%( a ), ( b ) \& ( c )>\n
Listing 5.
\% \OopsOptions
\%
\{% Inner = {(#1)},
\%
\Separators = {\{-\&-\textgreater\}},
\%
\Outer = {\textless#1\textgreater\}
\%
\}
\% \OopsNew{Foo}\{\textless ( a ), ( b ), Qux = ( c )\}
\%
%( a ), ( b ) \& ( c )>
We call $\omega_1, \ldots, \omega_n$ the elementary events, and

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

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

\begin{theorem}[Mittelwertsatz für $n$ Variable] Es sei
\end{theorem}
Dann gibt es auf jeder Strecke $\Strecke\subset\OffeneMenge$ einen Punkt $\xi\in\Strecke$, so dass gilt
\[
\frac{y\text{Differenz} = f(x) - f(x_0)}{x\text{Differenz} = x - x_0} = \text{Steigung} = \operatorname{grad} f(\xi)^\top
\]

Theorem 1 (Mittelwertsatz für $n$ Variable) Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{N}^n$ 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
\[
\frac{f(x) - f(x_0)}{x - x_0} = \text{grad} f(\xi)^\top
\]

Part III

Other

1 Acknowledgment

This work has benefited from Q&A’s from the \LaTeX{}community. For specifics, see here: https://tex.stackexchange.com/users/112708/erwann?tab=questions Listing 6 and Listing 7 are from \cite{1}. Listing 8 is from \texttt{tcolorbox}[4, 17.3].

References

[2] The \LaTeX{}X3 Project Team \textit{The \LaTeX{}X3 interfaces} \url{http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf}
[5] \url{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 \texttt{G} is supported but not recommended by \texttt{xparse}[3], but it’s really practical in this case.

4 Support

This package is available from \url{https://www.ctan.org/pkg/oops} (release) or \url{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

\NeedsTeXFormat{LaTeX2e}[2019/10/01]
\ExplSyntaxOn
\msg_new:nnn { Oops } { generic } { #1 }

1.1.2 Variables

\seq_new:N \__erw_oops_seq

1.1.3 Options

\keys_define:nn { Oops } {
  GenericObject .tl_gset:N = \__erw_oops_object_default_tl,
  GenericObject .value_required:n = false,
  GenericObject .default:n = {Math},
  GenericObject .initial:n = {Math},
  Inner .code:n =
  \cs_gset:Npn \__erw_oops_inner_default:n ##1{#1},
  Inner .value_required:n = false,
  Inner .default:n = {#1},
  Inner .initial:n = {#1},
  Separators .tl_gset:N = \__erw_oops_separators_default_tl,
  Separators .value_required:n = false,
  Separators .default:n = {{ \text{-and-}}{\text{,~}}{\text{,~and-}}},
  Separators .initial:n = {{ \text{-and-}}{\text{,~}}{\text{,~and-}}},
  Outer .code:n =
  \cs_gset:Npn \__erw_oops_outer_default:n #1{#1}
}

1.2 Prop

1.2.1 name

\cs_new:Npn \__erw_oops_name:n #1{\__erw_oops_#1}

1.2.2 new
\cs_new_protected:Nn \__erw_oops_new:n
  { \prop_new:c{\__erw_oops_name:n { #1 }} }
\cs_new_protected:Nn \__erw_oops_clear_new:n
  { \prop_clear_new:c{\__erw_oops_name:n { #1 }} }

1.2.3 put
\cs_new_protected:Nn \__erw_oops_put:nnn
  { \prop_put:cnn { \__erw_oops_name:n { #1 } } { #2 } { #3 } }
\cs_new_protected:Nn \__erw_oops_putinner:nnn
  { \__erw_oops_put:nnn{#1}{#2}{\__erw_oops_inner:n{#3}} }

1.2.4 item
\cs_new:Nn \__erw_oops_item:nn
  { \prop_item:cn { \__erw_oops_name:n { #1 } } { #2 } }
\cs_new:Npn \__erw_oops_item:nnn #1 #2 #3
  { \__erw_oops_item:nn{#1}{#2} }
\cs_new:Nn \__erw_oops_itemcmd:Nn
  { \ProvideDocumentCommand{#1}{O{\__erw_oops_object_default_tl}}
    { \__erw_oops_item:nn {##1}{ #2 } } }
\cs_generate_variant:Nn \__erw_oops_itemcmd:Nn {c}
\cs_new:Npn \__erw_oops_itemto_seq:nnn #1 #2 #3
  { \seq_put_right:Nn \__erw_oops_seq
    { \__erw_oops_item:nn{#1}{#2}} }

1.2.5 conditional
1.2.6 parse
\cs_set:Npn \__erw_oops_parse:Nnn
#1 % fun
#2 % prop name
#3 % clist
{
\tl_if_blank:nTF{#3}
{\c_empty_tl}
{
\seq_set_split:Nnn \l_tmpa_seq {,}{#3}
\cs_set:Npn \__erw_oops_parse:w ##1 = ##2 \q_stop
{#1}{#2}{\tl_trim_spaces:n{##1}} % key
{\tl_trim_spaces:n{##2}} % value
}\cs_set:Npn \__erw_oops_parse:n ##1
{\__erw_oops_parse:w ##1 \q_stop}
\seq_map_function:NN \l_tmpa_seq \__erw_oops_parse:n
}
}

2 Front end

2.1 \OopsOptions
\NewDocumentCommand{\OopsOptions}
{m}
{\keys_set:nn {Oops} {#1}}% \ProcessKeysPackageOptions{Oops}

2.2 \OopsClear
\NewDocumentCommand{\OopsClear}
{m}
{\__erw_oops_clear_new:n{#1}}

2.3 \OopsNew
\NewDocumentCommand{\OopsNew}
{m+o}
E{iso}
{
\__erw_oops_inner_default:n(#1)
\__erw_oops_separators_default_t1
\__erw_oops_outer_default:n(#1)
}
m
E{i}
{
\__erw_oops_inner_default:n(#1)
}
g
+o
{
\__erw_oops_if_exist:nTF{#1}
{\c_empty_tl}
{\__erw_oops_new:n{#1}}
\IfValueTF{#2}
{#2}
{\c_empty_tl}
\cs_gset:Npn
\__erw_oops_if_exist:nTF{#1}
\{#3\}
\__erw_oops_parse:Nnn
\__erw_oops_putinner:nnn
{#1}
{#6}
\__erw_oops_parse:Nnn
\__erw_oops_itemcmd:nnn
{#1}
{#6}
\IfValueTF{#2}
{
\seq_clear:N \__erw_oops_seq
\__erw_oops_parse:Nnn
\__erw_oops_itemto_seq:nnn
{#1}
{#6}
\cs_gset:Npn
\__erw_oops_outer:n #1
{#5}
\__erw_oops_outer:n
\exp_last_unbraced:NNo
\seq_use:Nnnn
\__erw_oops_seq
{#4}
}
\{#2\}
{\c_empty_tl}
\IfValueTF{#8}
{\IfValueTF{#9}
{\exp_not:n\OopsNew{\#1}{\#7}{\#8}{\#9}}
}{\exp_not:n\OopsNew{\#1}{\#7}{\#8}}
}
\IfValueTF{#9}
{\exp_not:n\OopsNew{\#1}{\#9}}
{\c_empty_tl}
\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.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>{(t13)}{(t14)}{(t15)} (option)</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Key&gt;</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>(\text{Key})</td>
<td></td>
<td>2, 6</td>
</tr>
<tr>
<td>(\text{kv1}) (option)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>(\text{code1})</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>(\text{code2})</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>(\text{Key})</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>(\text{code1}) (option)</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>(\text{code2}) (option)</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>(\text{code3}) (option)</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>(\text{kv11}) (option)</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>(\text{kv12}) (option)</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>(\text{t11}) (option)</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>(\text{t12}) (option)</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>(\text{t16}) (option)</td>
<td></td>
<td>6</td>
</tr>
</tbody>
</table>

C
\text{\texttt{\textbackslash code}} ................. 3

\text{cs commands:}
\text{\texttt{\textbackslash cs\_generate\_variant:NN}} ........ 68
\text{\texttt{\textbackslash cs\_get:NN}} .............. 16, 27, 147, 168
\text{\texttt{\textbackslash cs\_new:NN}} ............. 52, 60, 82
\text{\texttt{\textbackslash cs\_new:NN}} ............. 33, 56, 69, 76
\text{\texttt{\textbackslash cs\_new\_protected:NN}} .... 34, 38, 42, 46
\text{\texttt{\textbackslash cs\_set:NN}} ............. 86, 96, 103

D
\text{\texttt{\textbackslash documentclass}} .......... 3, 4