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

Erwann Rogard†

Released 2020/03/11

Abstract

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 *inline* thanks to a minimalist interface built upon xparse[3]. To make a definition, use `\OopsNew{`⟨*tl1*⟩`}`, where ⟨*tl1*⟩ 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 ⟨*tl1*⟩ 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

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

†firstname dot lastname AusTria gmail dot com

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 ⟨*tl*⟩, ⟨*kvl*⟩, ⟨*char*⟩, and ⟨*code*⟩.

c) `{`⟨*arg*⟩`}` is either `m` or `g` options[3], and `[`⟨*arg*⟩`]` and ⟨*char*⟩`{`⟨*arg*⟩`}` are `o` and `e` options[3].

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

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

f) We call `\code{`⟨*arg*⟩`}`, ⟨*code*⟩ with `#1` replaced by `{`⟨*arg*⟩`}`

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 ⟨*tl1*⟩ be a *token list* is ommited.

`\usepackage` `\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` `\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`, ⟨*tl1*⟩

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` `\OopsClear{`⟨*tl1*⟩`}`

Side effect Clears ⟨*tl1*⟩ of any ⟨*data*⟩

\OopsNew

```
\OopsNew{⟨tl1⟩}
[⟨tl2⟩]
i{⟨code1⟩}
s{{⟨tl3⟩}{⟨tl4⟩}{⟨tl5⟩}}
o{⟨code2⟩}
{⟨kvl1⟩}
i{⟨code3⟩}
{⟨kvl2⟩}
[⟨tl6⟩]
```

Requirement Only ⟨*tl1*⟩ and ⟨*kvl1*⟩ are mandatory

⟨*tl1*⟩

Use it to Identify an object

Side effect Registers ⟨*tl1*⟩ as a new object, if applicable

Example `Math`, `ModelA`, `ModelB`

⟨*tl2*⟩

Use it to Bring about a definition

Expands to ⟨*tl2*⟩

Example `Let~`

⟨*code1*⟩

Use it to Override `Inner`.

Example `\mathbb{#1}`

{⟨*tl3*⟩}{⟨*tl4*⟩}{⟨*tl5*⟩}

Use it to Override `Separators`

Example `{~\&~}{,~}{~\&~}`

⟨*code2*⟩

Use it to Override `Outer`

Example `\text{#1}`

⟨*kvl1*⟩

Side effect If `Key` is a new key, attaches to it ⟨*data*⟩=\⟨*code1*⟩`{Value}`

Expands to If ⟨*tl2*⟩ is *no value*, none, otherwise,

1. For each `Key`, calls `\Key`⟨*tl1*⟩. Call it ⟨*seq*⟩.

2. Concatenates ⟨*seq*⟩ using {⟨*tl3*⟩}{⟨*tl4*⟩}{⟨*tl5*⟩}. Call it ⟨*tmp*⟩.
3. Expands \⟨*code2*⟩{⟨*tmp*⟩}

Example Sample=\Omega

The remaining options taken together forward to:
\OopsNew{⟨*tl1*⟩} o{⟨*code3*⟩} {⟨*kvl2*⟩} [⟨*tl6*⟩]

⟨*code3*⟩
⟨*kvl2*⟩
⟨*tl6*⟩

\<Key> \⟨*Key*⟩[⟨*tl1*⟩]

Side effect Expands to ⟨*data*⟩ associated with ⟨*tl1*⟩ and ⟨*Key*⟩, if applicable.

Part II
Listings

Listing 1.

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

Listing 2.

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

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

Listing 3.

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

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

Listing 4.

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

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

Listing 5.

```
%     \OopsOptions
%     {
```

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

Listing 6.

```
%     \OopsNew{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.

```
%     \OopsClear{Math}
%     \OopsNew{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}
%     \OopsNew{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]}
%^^A     %Strecke={\char`[x_0,x\char`]}     % Also works
%^^A     %Strecke={\[x_0,x\]} % BUG
% }
%   [$n\in\N$, $\OffeneMenge\subseteq\N^n$ eine offene Menge und $f\in
    \Ci(\OffeneMenge,\R)$.
```

```
%   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 $n \in N$, $D \subseteq N^n$ eine offene Menge und $f \in C^1(D, R)$. 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 LaTeXcommunity. 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 tcolbox[4, 17.3].

References

[1] A.N. Shiryaev *Probability* Springer, 1995

[2] The LaTeX3 Project Team *The LaTeX3 interfaces* http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf

[3] The LaTeX3 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

```
\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}
  },
  Outer .value_required:n = false,
  Outer .default:n = {\ensuremath{#1}},
  Outer .initial:n = {\ensuremath{#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_itemcmd:nnn
#1
#2
#3
{
  \__erw_oops_itemcmd:cn{#2}{#2}
}
\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

```
\cs_new:Nn \__erw_oops_if_exist:nTF
{
  \prop_if_exist:cTF{\__erw_oops_name:n { #1 }}{#2}{#3}
}
```

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_tl}
    {\__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_inner:n ##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}
    }
  }
  {\c_empty_tl}
```

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