

Method Attributes

:Restricted
:Private
:Public

:ID
:PreInit
:Init
:Init(private | restricted | public)

:Destroy

:Automethod
name of method: \$_ (w/o classname)
real \$_: \$CALLER::\$_

:Replicate

:Dumper
:Pumper

:MOD_*_ATTRS
:FETCH_*_ATTRS

:Cumulative

:Chained
:Chained(top down) # default
:Chained(bottom up)

:MergeArgs

Introspection

methods can also be called on the class

```
$meta = $obj->meta  
  
@classes = $obj->isa  
@classes = $meta->get_classes  
  
@methods = $obj->can  
%methods = $meta->get_methods
```

Object::InsideOut Quick Reference

covers version 3.45

Coexion

:Stringify
:Numerify
:Boolify
:Arrayify
:Hashify
:Globify
:Codify

Field Attributes

attributes are not case-sensitive

:Field

:Default(\$value)

syn. for Default: Def

:Accessor(\$name)

:Accessor(Name => \$name,

Return => [old | new | object])

syn. for Accessor: Acc, Get_Set,

Combined, Combo

:Arg(\$name)

:Arg(name => \$name, Mandatory => \$bool,

Regex => \$re, Preprocess => \&sub)

syn. for Mandatory: Mand, Required,
Req

syn. for Regex: Re, Regexp

syn. for Preprocess: Pre, Preproc

:Get(\$name)

:Get(Name => \$name, %accessor_permission)

:Set(\$name)

:Set(Name => \$name,

Return => [old | new | object],

%accessor_permission)

syn. for Return: Ret

syn. for Old: Previous, Prev, Prior

:Std(\$name, %accessor_permission)

shortcut for :Set(\$name) :Get(\$name)

:Lvalue(Name => \$name,

Return => [old | new | object])

:All(\$name)

shorthand for :Arg(\$name) :Acc(\$name)

:Std_All(\$name)

:Std_All(Name => \$name, <set parameters>,

%accessor_permission)

like :All, but with :Std

:LV_All(\$name)

like :All, but with :Lvalue

:Type([scalar | numeric | list | list(_subtype_) |

classname |

ARRAY_ref | ARRAY_ref(_subtype) |

HASH | HASH_ref | ref_type |

\&My::Type::validate])

syn. for List: Array

:Deep

:Weak

%accessor_permission = (

Permission => [private | restricted | public],

Private => 1,

restricted => 1,

)

syn. for Permission: Perm

syn. for Private: Priv

syn. for Restricted: Rest

:InitArgs

```
my %init_args :InitArgs = (
```

```
  foo => "",
```

```
  bar => {
```

```
    Default => $value,
```

```
    Regex => qr/./,
```

```
    Type => $type,
```

```
    Preprocess => \&sub,
```

```
  },
```

```
);
```

If modified during runtime:

```
  Object::InsideOut::normalize(  
  \%init_args );
```

OIO Methods and Functions

```
$obj->set( \@my_field, $data )
```

valid only inside the class

```
$obj->dump( $as_string )
```

```
$obj->pump( $data )
```

```
$obj->clone( $deep )
```

```
My::Class->create_field( '@my_field',  
  @attributes )
```

```
My::Class->add_class( $class )
```