decorateme

Release 0.3.0

Douglas Myers-Turnbull

Aug 12, 2022

CONTENTS

1 API Reference 1

Python Module Index 17

Index 19

CHAPTER

ONE

API REFERENCE

This page contains auto-generated API reference documentation[1].

1.1 decorateme

Metadata for decorateme.

1.1.1 Submodules

decorateme._auto

Decorators for adding dunder methods automatically.

Module Contents

Functions

auto_utils()	Auto-adds __repr__, __str__, etc., for simple utility classes with no attributes.
auto_obj()	Auto-adds `__eq__`, `__hash__`, `__repr__`, `__str__`, and `_repr_html`.
auto_eq([only, exclude])	Auto-adds a __eq__ function by comparing its attributes.
auto_hash([only, exclude])	Auto-adds a __hash__ function by hashing its attributes.
auto_repr([only, exclude])	Auto-adds __repr__ and __str__.
auto_str([only, exclude, with_address])	Auto-adds `__str__`.
auto_html([only, exclude, with_address])	Auto-adds a `_repr_html` method, which Jupyter will use.
auto_repr_str([exclude_simple, exclude_html, exclude_all])	Decorator.
auto_info([only, exclude])	Auto-adds a function `info` that outputs a pretty multi-line representation of the instance and its attributes.

decorateme._auto.**auto_utils**()

Auto-adds __repr__, __str__, etc., for simple utility classes with no attributes.

[1] Created with sphinx-autoapi

decorateme._auto.**auto_obj**()

Auto-adds `__eq__`, `__hash__`, `__repr__`, `__str__`, and `_repr_html`. See the decorators for auto_eq, auto_hash, and auto_repr for more details.

decorateme._auto.**auto_eq**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = None*)

Auto-adds a __eq__ function by comparing its attributes.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes

decorateme._auto.**auto_hash**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = None*)

Auto-adds a __hash__ function by hashing its attributes.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes

decorateme._auto.**auto_repr**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = lambda a: ...*)

Auto-adds __repr__ and __str__.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes

decorateme._auto.**auto_str**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = lambda a: ...*, *with_address: bool = False*)

Auto-adds `__str__`.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes
- **with_address** – Include the hex memory address

decorateme._auto.**auto_html**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = lambda a: ...*, *with_address: bool = True*)

Auto-adds a `_repr_html` method, which Jupyter will use.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes
- **with_address** – Include the hex memory address

decorateme._auto.**auto_repr_str**(*exclude_simple: Optional[Callable[[str], bool]] = lambda a: ...*, *exclude_html: Optional[Callable[[str], bool]] = lambda a: ...*, *exclude_all: Optional[Callable[[str], bool]] = lambda a: ...*)

Decorator. Auto-adds `__repr__`, `__str__`, and `_repr_html_` that show the attributes:

- `__str__` will include attributes in neither `exclude_all` nor `exclude_simple`

- **_repr_html_ will include attributes in neither exclude_all nor exclude_simple**
  and will show the hexadecimal address
- __repr__ will include attributes not in exclude_all and will show the hexadecimal address

The _repr_html_ will be used by Jupyter display.

Example

```
repr(point) == Point(angle=0.3, radius=4, _style='point' @ 0x5528ca3)
str(point) == Point(angle=0.3, radius=4)
_repr_html_(point) == Point(angle=0.3, radius=4 @ 0x5528ca3)
```

Parameters

- **exclude_simple** – Exclude attributes matching these names in human-readable strings (str and _repr_html)
- **exclude_html** – Exclude for _repr_html
- **exclude_all** – Exclude these attributes in all the functions

decorateme._auto.**auto_info**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = lambda a: ...*)

Auto-adds a function `info` that outputs a pretty multi-line representation of the instance and its attributes.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes

decorateme._behavior

Decorators that affect object or class behavior.

Module Contents

Functions

takes_seconds_named(x, *args, **kwargs)	Prints a statement like "Call to calc_distances took 15.2s." after the function returns.
takes_seconds(x, *args, **kwargs)	Prints a statement like "Done. Took 15.2s." after the function returns.
mutable(cls)	Just marks an object as mutable.
immutable(mutableclass)	Decorator for making a slot-based class immutable.
auto_singleton(cls)	Makes it so the constructor returns a singleton instance.

decorateme._behavior.**takes_seconds_named**(*x*, **args*, ***kwargs*)

Prints a statement like “Call to calc_distances took 15.2s.” after the function returns.

decorateme._behavior.**takes_seconds**(*x*, **args*, ***kwargs*)

Prints a statement like “Done. Took 15.2s.” after the function returns.

decorateme._behavior.**mutable**(*cls*)

Just marks an object as mutable.

decorateme._behavior.**immutable**(*mutableclass*)

Decorator for making a slot-based class immutable. Taken almost verbatim from https://code.activestate.com/recipes/578233-immutable-class-decorator/ Written by Oren Tirosh and released under the MIT license.

decorateme._behavior.**auto_singleton**(*cls*)

Makes it so the constructor returns a singleton instance. The constructor CANNOT take arguments.

Example

```
@auto_singleton
class MyClass: pass
mysingleton = MyClass()
```

decorateme._doc

Decorators for manipulating docstrings.

Module Contents

Functions

copy_docstring(→ None)	Copies the docstring from *from_obj* to this function or class.
append_docstring(→ None)	Appends the docstring from *from_obj* to the docstring for this function or class.

decorateme._doc.**copy_docstring**(*from_obj: Type*) → None

Copies the docstring from *from_obj* to this function or class.

decorateme._doc.**append_docstring**(*from_obj: Type*) → None

Appends the docstring from *from_obj* to the docstring for this function or class.

decorateme._informative

Decorators that do nothing.

Module Contents

Functions

overrides(→ T)	Overriding this class is generally recommended (but not required).
override_recommended(→ T)	Overriding this class is suggested.
internal(→ T)	This class or package is meant to be used only by code within this project.
external(→ T)	This class or package is meant to be used *only* by code outside this project.
reserved(→ T)	This package, class, or function is empty but is declared for future use.
thread_safe(→ T)	Just marks that something **is** thread-safe.
not_thread_safe(→ T)	Just marks that something is **not** thread-safe.
recommend_final(→ T)	Marks as "should not override".

Attributes

T

decorateme._informative.**T**

decorateme._informative.**overrides**(*cls: T*) → T

Overriding this class is generally recommended (but not required).

decorateme._informative.**override_recommended**(*cls: T*) → T

Overriding this class is suggested.

decorateme._informative.**internal**(*cls: T*) → T

This class or package is meant to be used only by code within this project.

decorateme._informative.**external**(*cls: T*) → T

This class or package is meant to be used *only* by code outside this project.

decorateme._informative.**reserved**(*cls: T*) → T

This package, class, or function is empty but is declared for future use.

decorateme._informative.**thread_safe**(*cls: T*) → T

Just marks that something **is** thread-safe.

decorateme._informative.**not_thread_safe**(*cls: T*) → T

Just marks that something is **not** thread-safe.

decorateme._informative.**recommend_final**(*cls: T*) → T

Marks as “should not override”.

decorateme._over

Decorators for types that are "basically" a simpler type.

Module Contents

Functions

float_type(attribute)	Auto-adds a `__float__` using the `__float__` of some attribute.
int_type(attribute)	Auto-adds an `__int__` using the `__int__` of some attribute.
iterable_over(attribute)	Auto-adds an `__iter__` over elements in an iterable attribute.
collection_over(attribute)	Auto-adds an `__iter__` and `__len__` over elements in a collection attribute.
sequence_over(attribute)	Auto-adds `__getitem__` and `__len__` over elements in an iterable attribute.

decorateme._over.**float_type**(*attribute: str*)

Auto-adds a `__float__` using the `__float__` of some attribute. Used to annotate a class as being "essentially an float".

Parameters
attribute – The name of the attribute of this class

decorateme._over.**int_type**(*attribute: str*)

Auto-adds an `__int__` using the `__int__` of some attribute. Used to annotate a class as being "essentially an integer".

Parameters
attribute – The name of the attribute of this class

decorateme._over.**iterable_over**(*attribute: str*)

Auto-adds an `__iter__` over elements in an iterable attribute. Used to annotate a class as being "essentially an iterable" over some elements.

Parameters
attribute – The name of the attribute of this class

decorateme._over.**collection_over**(*attribute: str*)

Auto-adds an `__iter__` and `__len__` over elements in a collection attribute. Used to annotate a class as being "essentially a collection" over some elements.

Parameters
attribute – The name of the attribute of this class

decorateme._over.**sequence_over**(*attribute: str*)

Auto-adds `__getitem__` and `__len__` over elements in an iterable attribute. Used to annotate a class as being "essentially a list" over some elements.

Parameters
attribute – The name of the attribute of this class

decorateme._status

Decorators that warn about code maturity.

Module Contents

Classes

CodeStatus	An enum for the quality/maturity of code,

Functions

status(level[, vr, msg])	Annotate code quality. Emits a warning if bad code is called.

exception decorateme._status.**CodeIncompleteError**

Bases: NotImplementedError

The code is not finished.

exception decorateme._status.**CodeRemovedError**

Bases: NotImplementedError

The code was removed.

exception decorateme._status.**PreviewWarning**

Bases: UserWarning

The code being called is a preview, unstable. or immature.

class decorateme._status.**CodeStatus**

Bases: enum.Enum

An enum for the quality/maturity of code, ranging from incomplete to deprecated.

INCOMPLETE

PREVIEW

STABLE = 0

PENDING_DEPRECATION = 1

DEPRECATED = 2

REMOVED = 3

classmethod of(*x: Union[int, str,* CodeStatus*]*) → *CodeStatus*

decorateme._status.**status**(*level: Union[int, str,* CodeStatus*]*, *vr: Optional[str] = ''*, *msg: Optional[str] = None*)

Annotate code quality. Emits a warning if bad code is called.

Parameters

- **level** – The quality / maturity
- **vr** – First version the status / warning applies to
- **msg** – Explanation and/or when it will be removed or completed

decorateme._utils

Utilities for decorateme.

Module Contents

Classes

_SpecialStr	A string that can be displayed with Jupyter with line breaks and tabs.
_InfoSpecialStr	A string that can be displayed with Jupyter with line breaks and tabs.
_Utils	

class decorateme._utils.**_SpecialStr**(*s: str*)

Bases: `str`

A string that can be displayed with Jupyter with line breaks and tabs.

__repr__()

Return repr(self).

__str__()

Return str(self).

_repr_html_()

class decorateme._utils.**_InfoSpecialStr**(*s: str*)

Bases: *_SpecialStr*

A string that can be displayed with Jupyter with line breaks and tabs.

_repr_html_()

class decorateme._utils.**_Utils**

gen_str(*only: Optional[Set[str]] = None*, *exclude: Optional[Callable[[str], bool]] = None*, *bold_surround: Callable[[str], str] = str*, *em_surround: Callable[[str], str] = str*, *delim: str = ', '*, *eq: str = '='*, *opening: str = '('*, *closing: str = ')'*, *with_address: bool = True*)

classmethod var_items(*obj*, *only*, *exclude*)

classmethod var_values(*obj*, *only*, *exclude*)

classmethod auto_hash(*self*, *only: Optional[Set[str]]*, *exclude: Optional[Callable[[str], bool]]*)

classmethod auto_eq(*self*, *other*, *only: Optional[Set[str]]*, *exclude: Optional[Callable[[str], bool]]*)

1.1.2 Package Contents

Classes

CodeStatus	An enum for the quality/maturity of code,

Functions

auto_eq([only, exclude])	Auto-adds a __eq__ function by comparing its attributes.
auto_hash([only, exclude])	Auto-adds a __hash__ function by hashing its attributes.
auto_html([only, exclude, with_address])	Auto-adds a `_repr_html` method, which Jupyter will use.
auto_info([only, exclude])	Auto-adds a function `info` that outputs a pretty multi-line representation of the instance and its attributes.
auto_obj()	Auto-adds `__eq__`, `__hash__`, `__repr__`, `__str__`, and `_repr_html`.
auto_repr([only, exclude])	Auto-adds __repr__ and __str__.
auto_repr_str([exclude_simple, exclude_html, exclude_all])	Decorator.
auto_str([only, exclude, with_address])	Auto-adds `__str__`.
auto_utils()	Auto-adds __repr__, __str__, etc., for simple utility classes with no attributes.
auto_singleton(cls)	Makes it so the constructor returns a singleton instance.
immutable(mutableclass)	Decorator for making a slot-based class immutable.
mutable(cls)	Just marks an object as mutable.
takes_seconds(x, *args, **kwargs)	Prints a statement like "Done. Took 15.2s." after the function returns.
takes_seconds_named(x, *args, **kwargs)	Prints a statement like "Call to calc_distances took 15.2s." after the function returns.
append_docstring(→ None)	Appends the docstring from *from_obj* to the docstring for this function or class.
copy_docstring(→ None)	Copies the docstring from *from_obj* to this function or class.
external(→ T)	This class or package is meant to be used *only* by code outside this project.
internal(→ T)	This class or package is meant to be used only by code within this project.
not_thread_safe(→ T)	Just marks that something is **not** thread-safe.
override_recommended(→ T)	Overriding this class is suggested.
overrides(→ T)	Overriding this class is generally recommended (but not required).
reserved(→ T)	This package, class, or function is empty but is declared for future use.
thread_safe(→ T)	Just marks that something **is** thread-safe.
collection_over(attribute)	Auto-adds an `__iter__` and `__len__` over elements in a collection attribute.
float_type(attribute)	Auto-adds a `__float__` using the `__float__` of some attribute.
int_type(attribute)	Auto-adds an `__int__` using the `__int__` of some attribute.
iterable_over(attribute)	Auto-adds an `__iter__` over elements in an iterable attribute.
sequence_over(attribute)	Auto-adds `__getitem__` and `__len__` over elements in an iterable attribute.
status(level[, vr, msg])	Annotate code quality. Emits a warning if bad code is called.

Attributes

pkg
logger
metadata
metadata

decorateme.**pkg**

decorateme.**logger**

decorateme.**metadata**

decorateme.**metadata**

decorateme.**auto_eq**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = None*)

Auto-adds a __eq__ function by comparing its attributes.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes

decorateme.**auto_hash**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = None*)

Auto-adds a __hash__ function by hashing its attributes.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes

decorateme.**auto_html**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = lambda a: ...*, *with_address: bool = True*)

Auto-adds a `_repr_html` method, which Jupyter will use.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes
- **with_address** – Include the hex memory address

decorateme.**auto_info**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = lambda a: ...*)

Auto-adds a function `info` that outputs a pretty multi-line representation of the instance and its attributes.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes

decorateme.**auto_obj**()

Auto-adds `__eq__`, `__hash__`, `__repr__`, `__str__`, and `_repr_html`. See the decorators for auto_eq, auto_hash, and auto_repr for more details.

decorateme.**auto_repr**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = lambda a: ...*)

Auto-adds __repr__ and __str__.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes

decorateme.**auto_repr_str**(*exclude_simple: Optional[Callable[[str], bool]] = lambda a: ...*, *exclude_html: Optional[Callable[[str], bool]] = lambda a: ...*, *exclude_all: Optional[Callable[[str], bool]] = lambda a: ...*)

Decorator. Auto-adds `__repr__`, `__str__`, and `_repr_html_` that show the attributes:

- `__str__` will include attributes in neither `exclude_all` nor `exclude_simple`
- **`_repr_html_` will include attributes in neither `exclude_all` nor `exclude_simple`** and will show the hexadecimal address
- `__repr__` will include attributes not in exclude_all and will show the hexadecimal address

The `_repr_html_` will be used by Jupyter display.

Example

```
repr(point) == Point(angle=0.3, radius=4, _style='point' @ 0x5528ca3)
str(point) == Point(angle=0.3, radius=4)
_repr_html_(point) == Point(angle=0.3, radius=4 @ 0x5528ca3)
```

Parameters

- **exclude_simple** – Exclude attributes matching these names in human-readable strings (str and _repr_html)
- **exclude_html** – Exclude for _repr_html
- **exclude_all** – Exclude these attributes in all the functions

decorateme.**auto_str**(*only: Optional[AbstractSet[str]] = None*, *exclude: Optional[Callable[[str], bool]] = lambda a: ...*, *with_address: bool = False*)

Auto-adds `__str__`.

Parameters

- **only** – Only include these attributes
- **exclude** – Exclude these attributes
- **with_address** – Include the hex memory address

decorateme.**auto_utils**()

Auto-adds __repr__, __str__, etc., for simple utility classes with no attributes.

decorateme.**auto_singleton**(*cls*)

Makes it so the constructor returns a singleton instance. The constructor CANNOT take arguments.

Example

```
@auto_singleton
class MyClass: pass
mysingleton = MyClass()
```

decorateme.**immutable**(*mutableclass*)

Decorator for making a slot-based class immutable. Taken almost verbatim from https://code.activestate.com/recipes/578233-immutable-class-decorator/ Written by Oren Tirosh and released under the MIT license.

decorateme.**mutable**(*cls*)

Just marks an object as mutable.

decorateme.**takes_seconds**(*x*, **args*, ***kwargs*)

Prints a statement like "Done. Took 15.2s." after the function returns.

decorateme.**takes_seconds_named**(*x*, **args*, ***kwargs*)

Prints a statement like "Call to calc_distances took 15.2s." after the function returns.

decorateme.**append_docstring**(*from_obj: Type*) → None

Appends the docstring from *from_obj* to the docstring for this function or class.

decorateme.**copy_docstring**(*from_obj: Type*) → None

Copies the docstring from *from_obj* to this function or class.

decorateme.**external**(*cls: T*) → T

This class or package is meant to be used *only* by code outside this project.

decorateme.**internal**(*cls: T*) → T

This class or package is meant to be used only by code within this project.

decorateme.**not_thread_safe**(*cls: T*) → T

Just marks that something is **not** thread-safe.

decorateme.**override_recommended**(*cls: T*) → T

Overriding this class is suggested.

decorateme.**overrides**(*cls: T*) → T

Overriding this class is generally recommended (but not required).

decorateme.**reserved**(*cls: T*) → T

This package, class, or function is empty but is declared for future use.

decorateme.**thread_safe**(*cls: T*) → T

Just marks that something **is** thread-safe.

decorateme.**collection_over**(*attribute: str*)

Auto-adds an `__iter__` and `__len__` over elements in a collection attribute. Used to annotate a class as being "essentially a collection" over some elements.

Parameters

attribute – The name of the attribute of this class

decorateme.**float_type**(*attribute: str*)

Auto-adds a `__float__` using the `__float__` of some attribute. Used to annotate a class as being "essentially an float".

Parameters
attribute – The name of the attribute of this class

decorateme.**int_type**(*attribute: str*)

Auto-adds an `__int__` using the `__int__` of some attribute. Used to annotate a class as being "essentially an integer".

Parameters
attribute – The name of the attribute of this class

decorateme.**iterable_over**(*attribute: str*)

Auto-adds an `__iter__` over elements in an iterable attribute. Used to annotate a class as being "essentially an iterable" over some elements.

Parameters
attribute – The name of the attribute of this class

decorateme.**sequence_over**(*attribute: str*)

Auto-adds `__getitem__` and `__len__` over elements in an iterable attribute. Used to annotate a class as being "essentially a list" over some elements.

Parameters
attribute – The name of the attribute of this class

exception decorateme.**CodeIncompleteError**

Bases: `NotImplementedError`

The code is not finished.

exception decorateme.**CodeRemovedError**

Bases: `NotImplementedError`

The code was removed.

class decorateme.**CodeStatus**

Bases: `enum.Enum`

An enum for the quality/maturity of code, ranging from incomplete to deprecated.

INCOMPLETE

PREVIEW

STABLE = 0

PENDING_DEPRECATION = 1

DEPRECATED = 2

REMOVED = 3

classmethod of(*x: Union[int, str,* CodeStatus*]*) → *CodeStatus*

exception decorateme.**PreviewWarning**

Bases: `UserWarning`

The code being called is a preview, unstable. or immature.

decorateme.**status**(*level: Union[int, str, CodeStatus]*, *vr: Optional[str] = ''*, *msg: Optional[str] = None*)

Annotate code quality. Emits a warning if bad code is called.

Parameters

- **level** – The quality / maturity
- **vr** – First version the status / warning applies to
- **msg** – Explanation and/or when it will be removed or completed

PYTHON MODULE INDEX

d

decorateme, 1
decorateme._auto, 1
decorateme._behavior, 3
decorateme._doc, 4
decorateme._informative, 4
decorateme._over, 6
decorateme._status, 7
decorateme._utils, 8

INDEX

Symbols

_InfoSpecialStr (*class in decorateme._utils*), 8
_SpecialStr (*class in decorateme._utils*), 8
_Utils (*class in decorateme._utils*), 8
__repr__() (*decorateme._utils._SpecialStr method*), 8
__str__() (*decorateme._utils._SpecialStr method*), 8
_repr_html_() (*decorateme._utils._InfoSpecialStr method*), 8
_repr_html_() (*decorateme._utils._SpecialStr method*), 8

A

append_docstring() (*in module decorateme*), 13
append_docstring() (*in module decorateme._doc*), 4
auto_eq() (*decorateme._utils._Utils class method*), 8
auto_eq() (*in module decorateme*), 11
auto_eq() (*in module decorateme._auto*), 2
auto_hash() (*decorateme._utils._Utils class method*), 8
auto_hash() (*in module decorateme*), 11
auto_hash() (*in module decorateme._auto*), 2
auto_html() (*in module decorateme*), 11
auto_html() (*in module decorateme._auto*), 2
auto_info() (*in module decorateme*), 11
auto_info() (*in module decorateme._auto*), 3
auto_obj() (*in module decorateme*), 11
auto_obj() (*in module decorateme._auto*), 1
auto_repr() (*in module decorateme*), 12
auto_repr() (*in module decorateme._auto*), 2
auto_repr_str() (*in module decorateme*), 12
auto_repr_str() (*in module decorateme._auto*), 2
auto_singleton() (*in module decorateme*), 12
auto_singleton() (*in module decorateme._behavior*), 4
auto_str() (*in module decorateme*), 12
auto_str() (*in module decorateme._auto*), 2
auto_utils() (*in module decorateme*), 12
auto_utils() (*in module decorateme._auto*), 1

C

CodeIncompleteError, 7, 14
CodeRemovedError, 7, 14
CodeStatus (*class in decorateme*), 14
CodeStatus (*class in decorateme._status*), 7
collection_over() (*in module decorateme*), 13
collection_over() (*in module decorateme._over*), 6
copy_docstring() (*in module decorateme*), 13
copy_docstring() (*in module decorateme._doc*), 4

D

decorateme
    module, 1
decorateme._auto
    module, 1
decorateme._behavior
    module, 3
decorateme._doc
    module, 4
decorateme._informative
    module, 4
decorateme._over
    module, 6
decorateme._status
    module, 7
decorateme._utils
    module, 8
DEPRECATED (*decorateme._status.CodeStatus attribute*), 7
DEPRECATED (*decorateme.CodeStatus attribute*), 14

E

external() (*in module decorateme*), 13
external() (*in module decorateme._informative*), 5

F

float_type() (*in module decorateme*), 13
float_type() (*in module decorateme._over*), 6

G

gen_str() (*decorateme._utils._Utils method*), 8

I

immutable() (*in module decorateme*), 13
immutable() (*in module decorateme._behavior*), 4

INCOMPLETE (*decorateme._status.CodeStatus attribute*), 7
INCOMPLETE (*decorateme.CodeStatus attribute*), 14
int_type() (*in module decorateme*), 14
int_type() (*in module decorateme._over*), 6
internal() (*in module decorateme*), 13
internal() (*in module decorateme._informative*), 5
iterable_over() (*in module decorateme*), 14
iterable_over() (*in module decorateme._over*), 6

L

logger (*in module decorateme*), 11

M

metadata (*in module decorateme*), 11
module
    decorateme, 1
    decorateme._auto, 1
    decorateme._behavior, 3
    decorateme._doc, 4
    decorateme._informative, 4
    decorateme._over, 6
    decorateme._status, 7
    decorateme._utils, 8
mutable() (*in module decorateme*), 13
mutable() (*in module decorateme._behavior*), 3

N

not_thread_safe() (*in module decorateme*), 13
not_thread_safe() (*in module decorateme._informative*), 5

O

of() (*decorateme._status.CodeStatus class method*), 7
of() (*decorateme.CodeStatus class method*), 14
override_recommended() (*in module decorateme*), 13
override_recommended() (*in module decorateme._informative*), 5
overrides() (*in module decorateme*), 13
overrides() (*in module decorateme._informative*), 5

P

PENDING_DEPRECATION (*decorateme._status.CodeStatus attribute*), 7
PENDING_DEPRECATION (*decorateme.CodeStatus attribute*), 14
pkg (*in module decorateme*), 11
PREVIEW (*decorateme._status.CodeStatus attribute*), 7
PREVIEW (*decorateme.CodeStatus attribute*), 14
PreviewWarning, 7, 14

R

recommend_final() (*in module decorateme._informative*), 5
REMOVED (*decorateme._status.CodeStatus attribute*), 7
REMOVED (*decorateme.CodeStatus attribute*), 14
reserved() (*in module decorateme*), 13
reserved() (*in module decorateme._informative*), 5

S

sequence_over() (*in module decorateme*), 14
sequence_over() (*in module decorateme._over*), 6
STABLE (*decorateme._status.CodeStatus attribute*), 7
STABLE (*decorateme.CodeStatus attribute*), 14
status() (*in module decorateme*), 14
status() (*in module decorateme._status*), 7

T

T (*in module decorateme._informative*), 5
takes_seconds() (*in module decorateme*), 13
takes_seconds() (*in module decorateme._behavior*), 3
takes_seconds_named() (*in module decorateme*), 13
takes_seconds_named() (*in module decorateme._behavior*), 3
thread_safe() (*in module decorateme*), 13
thread_safe() (*in module decorateme._informative*), 5

V

var_items() (*decorateme._utils._Utils class method*), 8
var_values() (*decorateme._utils._Utils class method*), 8