
strong

Jan 17, 2021

1	Quickstart	3
2	Contributor Guide	7
3	Indices and tables	9
	Python Module Index	11
	Index	13



1.1 Installation

To install **Strong**, simply using pip:

```
python<version> -m pip install strong
```

Note: This package mostly relies on the quite recent **inspect** and **typing** package features. This package therefore require a Python installation with version ≥ 3.8 .

1.2 Read the documentation

1.2.1 Core package

Core modules

`core.decorators` module

`core.signature` module

1.2.2 Utils package

Utils

utils.output module

`utils.output.DEFAULT_OUTPUT(msg: str) → None`

Prints a message in the error output.

Parameters `msg` – the message

Example

```
>>> raise_stderr("Hello")
Hello # in red
```

`utils.output.raise_assertion_error(msg: str) → None`

Raises an assertion error with given message.

Parameters `msg` – the message

Raises `AssertionError`

Example

```
>>> raise_assertion_error("Hello")
AssertionError: Hello
```

`utils.output.raise_stdout(msg: str) → None`

Prints a message in the error output.

Parameters `msg` – the message

Example

```
>>> raise_stdout("Hello")
Hello # in red
```

`utils.output.raise_stdout(msg: str) → None`

Prints a message in the standard output.

Parameters `msg` – the message

Example

```
>>> raise_stdout("Hello")
Hello
```

`utils.output.raise_type_error(msg: str) → None`

Raises a type error with given message.

Parameters `msg` – the message

Raises `TypeError`

Example

```
>>> raise_type_error("Hello")
TypeError: Hello
```

`utils.output.raise_warning(msg: str) → None`

Raises a warning with given message.

Parameters `msg` – the message

Example


```
>>> raise_warning("Hello")  
Warning: Hello
```


CHAPTER 2

Contributor Guide

If you want to help in any way, feel free to check the [Github repository](#). This package follows the same coding guidelines as the one used by default with the Flake8 tool:

```
flake8 strong
flake8 tests
```

And Strong provide additional checks on typing:

```
strong strong
```

Code reformatting is done using Black tool:

```
black strong -l 79
black tests -l 79
```

Lastly, make sure to run the tests:

```
pytest tests
```


CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

u

`utils.output`, 4

D

`DEFAULT_OUTPUT()` (*in module `utils.output`*), 4

R

`raise_assertion_error()` (*in module `utils.output`*), 4

`raise_stderr()` (*in module `utils.output`*), 4

`raise_stdout()` (*in module `utils.output`*), 4

`raise_type_error()` (*in module `utils.output`*), 4

`raise_warning()` (*in module `utils.output`*), 4

U

`utils.output` (*module*), 4