
python*testingadvancedDocumentation*

Release 0.3.6

Julio Antúnez Tarín

Feb 23, 2019

Contents:

1	Python Testing Advanced	1
1.1	Information	1
1.2	Credits	1
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
3.1	Doctest	5
4	python_testing_advanced	7
4.1	python_testing_advanced package	7
5	Features	9
6	Contributing	11
6.1	Types of Contributions	11
6.2	Get Started!	12
6.3	Pull Request Guidelines	13
6.4	Tips	13
6.5	Deploying	13
7	Credits	15
7.1	Development Lead	15
7.2	Contributors	15
8	Changelog	17
8.1	v0.3.6 (2018-08-02)	17
8.2	v0.3.5 (2018-08-02)	17
8.3	v0.3.4 (2018-07-31)	17
8.4	v0.3.3 (2018-07-31)	17
8.5	v0.3.2 (2018-07-31)	18
8.6	v0.3.1 (2018-07-31)	18
8.7	v0.3.0 (2018-07-31)	18
8.8	v0.2.1 (2018-07-31)	18
8.9	v0.2.0 (2018-07-31)	18

8.10	v0.1.4 (2018-07-31)	18
8.11	v0.1.3 (2018-07-31)	18
8.12	v0.1.2 (2018-07-31)	19
8.13	v0.1.1 (2018-07-31)	19
8.14	v0.1.0 (2018-07-31)	19
9	Indices and tables	21
	Python Module Index	23

1.1 Information

Python Testing Advanced Example

- Free software: [MIT license](#)
- Documentation: <https://python-testing-advanced.readthedocs.io>.

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install Python Testing Advanced, run this command in your terminal:

```
$ pip install python_testing_advanced
```

This is the preferred method to install Python Testing Advanced, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for Python Testing Advanced can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/jatap/python_testing_advanced
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/jatap/python_testing_advanced/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

Usage

To use Python Testing Advanced in a project:

```
from python_testing_advanced import python_testing_advanced
```

3.1 Doctest

Note: Sphinx docs for `doctest`,

Example using `doctest` :

```
>> print(python_testing_advanced.help())  
Python Testing Advanced Tutorial Module
```

Example using `testcode` :

```
print(python_testing_advanced.help())
```

Example using `testoutput` :

```
Python Testing Advanced Tutorial Module
```

python_testing_advanced

4.1 python_testing_advanced package

4.1.1 Submodules

4.1.2 python_testing_advanced.python_testing_advanced module

Main module

Note: Proof of Concept

Usage: from python_testing_advanced import python_testing_advanced

```
python_testing_advanced.python_testing_advanced.help()  
Python Testing Advanced Module help
```

Todo: Add some subpackage

Args: none

Returns: A brief description of the module

```
python_testing_advanced.python_testing_advanced.init()  
Print help() when run as a script.
```

Args: none

Returns: `python_testing_advanced.python_testing_advanced.help()`

4.1.3 Module contents

Top-level package for Python Testing Advanced.

CHAPTER 5

Features

Todo: Add some subpackage

original entry

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at https://github.com/jatap/python_testing_advanced/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

6.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

6.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

6.1.4 Write Documentation

Python Testing Advanced could always use more documentation, whether as part of the official Python Testing Advanced docs, in docstrings, or even on the web in blog posts, articles, and such.

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/jatap/python_testing_advanced/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

6.2 Get Started!

Ready to contribute? Here's how to set up *python_testing_advanced* for local development.

1. Fork the *python_testing_advanced* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/python_testing_advanced.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv python_testing_advanced
$ cd python_testing_advanced/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 python_testing_advanced tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

6.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/jatap/python_testing_advanced/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

```
$ py.test tests.test_python_testing_advanced
```

6.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER 7

Credits

```
master_doc = 'index'
```

7.1 Development Lead

- Julio Antúnez Tarín <julio.antunez.tarin@gmail.com>

7.2 Contributors

None yet. Why not be the first?

8.1 v0.3.6 (2018-08-02)

8.1.1 Documentation Changes

- [tests] rewrite docstrings

8.2 v0.3.5 (2018-08-02)

8.2.1 Administration and Chores

- [packages] update requirements_dev file with bare ones

8.3 v0.3.4 (2018-07-31)

8.3.1 Documentation Changes

- [rst] move CHANGELOG.md to HISTORY.rst

8.4 v0.3.3 (2018-07-31)

8.4.1 Administration and Chores

- [files] delete deprecated HISTORY.md

8.5 v0.3.2 (2018-07-31)

8.5.1 Administration and Chores

- [pyup] update packages

8.6 v0.3.1 (2018-07-31)

8.6.1 Administration and Chores

- [pyup] update packages

8.7 v0.3.0 (2018-07-31)

8.7.1 New Features

- [packages] add Markdown support

8.8 v0.2.1 (2018-07-31)

8.8.1 Documentation Changes

- [modules] refactor main module docstring

8.9 v0.2.0 (2018-07-31)

8.9.1 New Features

- [modules] refactor main module and its tests

8.10 v0.1.4 (2018-07-31)

8.11 v0.1.3 (2018-07-31)

8.11.1 Documentation Changes

- [rst] update rst files

8.12 v0.1.2 (2018-07-31)

8.12.1 Documentation Changes

- [sphinx] update theme

8.13 v0.1.1 (2018-07-31)

8.13.1 Documentation Changes

- [readme] update

8.14 v0.1.0 (2018-07-31)

8.14.1 Administration and Chores

- [init] first commit

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`

p

`python_testing_advanced`, [8](#)
`python_testing_advanced.python_testing_advanced`
 (*OS X*), [7](#)

H

`help()` (*in module `python_testing_advanced.python_testing_advanced`*),
[7](#)

I

`init()` (*in module `python_testing_advanced.python_testing_advanced`*),
[7](#)

P

`python_testing_advanced(module)`, [8](#)
`python_testing_advanced.python_testing_advanced`
`(module)`, [7](#)