

---

# **Irritable Documentation**

***Release 0.1.0***

**Tim Bell**

**Aug 08, 2017**



---

## Contents

---

<b>1</b>	<b>Irritable</b>	<b>3</b>
1.1	Features . . . . .	3
1.2	Credits . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Stable release . . . . .	5
2.2	From sources . . . . .	5
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	11
4.4	Tips . . . . .	11
<b>5</b>	<b>Credits</b>	<b>13</b>
5.1	Development Lead . . . . .	13
5.2	Contributors . . . . .	13
<b>6</b>	<b>History</b>	<b>15</b>
6.1	0.1.0 (2017-08-08) . . . . .	15
<b>7</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



Irritable implements broken iterators called irritables

- Free software: BSD license
- Documentation: <https://irritable.readthedocs.io>.

## Features

Iterators are defined in the [docs](#); the definition includes what constitutes a broken implementation:

### 4.5. Iterator Types

[...]

Once an iterator's `__next__()` method raises `StopIteration`, it must continue to do so on subsequent calls. Implementations that do not obey this property are deemed broken.

*Irritables* are like iterables, but deliberately broken. The following types of brokenness are supported:

- after `next()` first raises `StopIteration`, subsequent calls will raise `StopIrritation` instead;
- if the irritator is instantiated with `resume=True`, calling `next()` may raise `StopIteration` when items still remain in the container; subsequent calls to `next()` will return the remaining items as usual until none remain;
- if the irritator is instantiated with `repeat=True`, after the iterator has been exhausted and raises `StopIteration`, the iterator is reset and can be iterated over again ad infinitum; in this case `next()` will never raise `StopIrritation`;

## Credits

The idea for irritators came during Trey Hunner's [talk](#) "Loop better: a deeper look at iteration in Python" at DjangoCon AU 2017.

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



### Stable release

To install Irritable, run this command in your terminal:

```
$ pip install irritable
```

This is the preferred method to install Irritable, 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.

### From sources

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

You can either clone the public repository:

```
$ git clone git://github.com/timb07/irritable
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/timb07/irritable/tarball/master
```

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

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



## CHAPTER 3

---

### Usage

---

To use Irritable in a project:

```
import irritable
```



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

### Types of Contributions

#### Report Bugs

Report bugs at <https://github.com/timb07/irritable/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.

#### Fix Bugs

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

#### Implement Features

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

## Write Documentation

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

## Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/timb07/irritable/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 :)

## Get Started!

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

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

```
$ git clone git@github.com:your_name_here/irritable.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 irritable
$ cd irritable/
$ 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 irritable 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.

## 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check [https://travis-ci.org/timb07/irritable/pull\\_requests](https://travis-ci.org/timb07/irritable/pull_requests) and make sure that the tests pass for all supported Python versions.

## Tips

To run a subset of tests:

```
$ py.test tests.test_irritable
```





## CHAPTER 5

---

### Credits

---

### Development Lead

- Tim Bell <[timothybell@gmail.com](mailto:timothybell@gmail.com)>

### Contributors

None yet. Why not be the first?



## CHAPTER 6

---

### History

---

#### **0.1.0 (2017-08-08)**

- First release on PyPI.



## CHAPTER 7

---

### Indices and tables

---

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