
Python BiDi Documentation

Release 0.4.0

Meir Kriheli

January 11, 2016

1	Installation	3
2	Usage	5
3	Contributing	7
3.1	Types of Contributions	7
3.2	Get Started!	8
3.3	Pull Request Guidelines	8
3.4	Tips	9
4	Credits	11
4.1	Development Lead	11
4.2	Tests	11
4.3	Contributors	11
5	History	13
5.1	0.4.0	13
5.2	0.3.4	13
5.3	0.3.3	13
5.4	0.3.2	13
5.5	0.3.1	14
5.6	0.3	14
5.7	0.2	14
5.8	0.1	14
6	Indices and tables	15

Bi-directional (BiDi) layout implementation in pure python.

Contents:

Installation

At the command line:

```
$ easy_install python-bidi
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv python-bidi  
$ pip install python-bidi
```

Usage

To use Python BiDi in a project:

```
.. code-block:: python
```

```
from bidi import algorithm
```

```
some_string = 'your string goes here' result = algorithm.get_display(some_string)
```

Contributing

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

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at <https://github.com/MeirKriheli/python-bidi/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.

3.1.2 Fix Bugs

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

3.1.3 Implement Features

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

3.1.4 Write Documentation

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

3.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/MeirKriheli/python-bidi/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 :)

3.2 Get Started!

Ready to contribute? Here's how to set up *python-bidi* for local development.

1. Fork the *python-bidi* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/python-bidi.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-bidi
$ cd python-bidi/
$ 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-bidi tests
$ python setup.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.

3.3 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 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/MeirKriheli/python-bidi/pull_requests and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_python-bidi
```

Credits

4.1 Development Lead

- Meir Kriheli <mkriheli@gmail.com>

4.2 Tests

Tests based on [fribidi](#).

4.3 Contributors

None yet. Why not be the first?

History

5.1 0.4.0

- Move to cookiecutter template
- Python 3 support (py2.6, 2.7, 3.3, 3.4 and pypy)
- Better docs
- Travis integration
- Tox tests
- PEP8 cleanup

5.2 0.3.4

- Remove extra newline in console script output

5.3 0.3.3

- Implement overriding base paragraph direction
- Allow overriding base direction in pybidi console script
- Fix returning display in same encoding

5.4 0.3.2

- Test for surrogate pairs
- Fix indentation in documentations
- Specify license in setup.py

5.5 0.3.1

- Added missing description
- docs/INSTALL.rst

5.6 0.3

- Apply bidi mirroring
- Move to back function based implementation

5.7 0.2

- Move the algorithm to a class based implementation

5.8 0.1

- Initial release

Indices and tables

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