

---

# **recursive***dictDocumentation*

***Release 0.2.0***

**Ryan Anguiano**

December 29, 2016



<b>1</b>	<b>recursive_dict</b>	<b>3</b>
1.1	Features . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
<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 . . . . .	10
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>
<b>7</b>	<b>0.2.0 (2016-12-29)</b>	<b>17</b>
<b>8</b>	<b>0.1.0 (2015-01-11)</b>	<b>19</b>
<b>9</b>	<b>Indices and tables</b>	<b>21</b>



Contents:



---

## recursive\_dict

---

A dict subclass that can lookup several layers deep.

- Free software: ISC license
- Documentation: [https://recursive\\_dict.readthedocs.org](https://recursive_dict.readthedocs.org).

### 1.1 Features

- TODO





---

# Installation

---

At the command line:

```
$ easy_install recursive_dict
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv recursive_dict  
$ pip install recursive_dict
```



---

### Usage

---

To use `recursive_dict` in a project:

```
import recursive_dict
```



---

## 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:

### 4.1 Types of Contributions

#### 4.1.1 Report Bugs

Report bugs at [https://github.com/ryananguiano/recursive\\_dict/issues](https://github.com/ryananguiano/recursive_dict/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.

#### 4.1.2 Fix Bugs

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

#### 4.1.3 Implement Features

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

#### 4.1.4 Write Documentation

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

#### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at [https://github.com/ryananguiano/recursive\\_dict/issues](https://github.com/ryananguiano/recursive_dict/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 :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *recursive\_dict* for local development.

1. Fork the *recursive\_dict* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:ryananguiano/recursive_dict.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 recursive_dict
$ cd recursive_dict/
$ 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 recursive_dict 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.

## 4.3 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/ryananguiano/recursive\\_dict/pull\\_requests](https://travis-ci.org/ryananguiano/recursive_dict/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_recursive_dict
```





---

**Credits**

---

## 5.1 Development Lead

- Ryan Anguiano <[ryan.anguiano@gmail.com](mailto:ryan.anguiano@gmail.com)>

## 5.2 Contributors

None yet. Why not be the first?



---

**History**

---



---

**0.2.0 (2016-12-29)**

---

- Add `__setitem__` method
- Create a base class
- Retain a link to original item
- Access original item attributes



---

**0.1.0 (2015-01-11)**

---

- First release on PyPI.





---

## Indices and tables

---

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