

---

# **TV Subs Rename Documentation**

*Release 0.1.0*

**Elias Dorneles**

November 15, 2014



<b>1</b>	<b>TV Subs Rename</b>	<b>3</b>
1.1	Usage . . . . .	3
1.2	How it works . . . . .	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>0.1.0 (2014-11-16)</b>	<b>17</b>
<b>8</b>	<b>Indices and tables</b>	<b>19</b>



Contents:



---

## TV Subs Rename

---

Script for mass-renaming TV subtitles for a given set of video files

- Free software: BSD license
- Documentation: [https://tvsubs\\_rename.readthedocs.org](https://tvsubs_rename.readthedocs.org).

### 1.1 Usage

```
tvsubs_rename *.avi
```

This will look for subtitle files matching the episode numbers for all the \*.avi files in current directory and rename them to match the video filenames.

### 1.2 How it works

It matches episode filenames looking for commonly used patterns in TV-series episode files (like s01e02 or 1x02 to indicate second episode of season 1) and will rename it to match the video filename.





---

## Installation

---

At the command line:

```
$ easy_install tvsubs_rename
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv tvsubs_rename  
$ pip install tvsubs_rename
```



---

**Usage**

---

To use TV Subs Rename in a project:

```
import tvsubs_rename
```



---

## 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/eliasdorneles/tvsubs\\_rename/issues](https://github.com/eliasdorneles/tvsubs_rename/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

TV Subs Rename could always use more documentation, whether as part of the official TV Subs Rename 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/eliasdorneles/tvsubs\\_rename/issues](https://github.com/eliasdorneles/tvsubs_rename/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 *tvsubs\_rename* for local development.

1. Fork the *tvsubs\_rename* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/tvsubs_rename.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 tvsubs_rename
$ cd tvsubs_rename/
$ 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 tvsubs_rename 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/eliasdorneles/tvsubs\\_rename/pull\\_requests](https://travis-ci.org/eliasdorneles/tvsubs_rename/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_tvsubs_rename
```





**Credits**

---

## 5.1 Development Lead

- Elias Dorneles <eliasdorneles@gmail.com>

## 5.2 Contributors

None yet. Why not be the first?



---

**History**

---



---

**0.1.0 (2014-11-16)**

---

- First release on PyPI.



---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*