Remarker Documentation

Release 0.0.9+5.g5bdc2dd.dirty

Dave Forgac

Apr 16, 2019

Contents

Contents:

CHAPTER 1

Remarker

A command line tool for generating Remark.js presentations from markdown files.

License: MIT

Documentation: https://remarker.readthedocs.org.

1.1 Usage

```
remarker --help
```

```
Usage: remarker [OPTIONS] SLIDES_MARKDOWN_FILE

  Generate a Remark.js HTML presentation from input Markdown and optional
  custom CSS.

Options:
  --version                   Show the version and exit.
  -v, --verbose               Output debugging info.
  -t, --title TEXT            HTML title of the presentation.
  -o, --output-file FILENAME  Write the output to a file instead of STDOUT.
  -c, --css-file PATH         Custom CSS to be included inline.
  --html-template PATH        Jinja2 template file for the presentation.
  --help                      Show this message and exit.
```

1.2 Usage Examples

Generate `presentation.html` from Markdown in `slides.md`:

```
remarker -o presentation.html slides.md
```

Generate `presentation.html` from Markdown in `slides.md` and CSS in `style.css`:

```
remarker -o presentation.html -c style.css slides.md
```

CHAPTER 2

Installation

Recommended: Install with `pipsi` in order to use `remarker` as a standalone command:

First, install pipsi.

Then:

```
$ pipsi install Remarker
```

Or, to install in your active Python environment:

```
$ pip install Remarker
```

CHAPTER 3

Usage

Run `remarker --help`

CHAPTER 4

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/tylerdave/remarker/issues.

If you are reporting a bug, please include:

- 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

Remarker could always use more documentation, whether as part of the official Remarker 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/tylerdave/remarker/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 *Remarker* for local development.

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

```
$ git clone git@github.com:your_name_here/remarker.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 remarker
$ cd remarker/
$ 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 remarker 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 2.7, 3.3, 3.4, 3.5 and for PyPy. Check https://travis-ci.org/tylerdave/remarker/pull_requests and make sure that the tests pass for all supported Python versions.

CHAPTER 5

Credits

5.1 Maintainer

- Dave Forgac <tylerdave@tylerdave.com>

5.2 Contributors

None yet. Why not be the first? See: CONTRIBUTING.rst

CHAPTER 6

History

Pre-release

CHAPTER 7

Indices and tables

- genindex
- modindex
- search