
CommonMark-py Documentation

Release 0.8.1

Roland Shoemaker, Bibek Kafle

Sep 10, 2018

Contents

1	CommonMark-py	1
1.1	Installation	1
1.2	Usage	1
1.3	Contributing	2
1.4	Tests	2
1.5	Authors	3

CHAPTER 1

CommonMark-py

CommonMark-py is a pure Python port of [jgm's commonmark.js](#), a Markdown parser and renderer for the [CommonMark](#) specification, using only native modules. Once both this project and the CommonMark specification are stable we will release the first 1.0 version and attempt to keep up to date with changes in `commonmark.js`.

CommonMark-py is tested against the CommonMark spec with Python versions 2.6, 2.7, 3.3, 3.4, 3.5, 3.6, and 3.7.

Current version: 0.8.1

1.1 Installation

```
$ pip install commonmark
```

1.2 Usage

```
>>> import commonmark
>>> commonmark.commonmark('*hello!')
'<p><em>hello!</em></p>\n'
```

Or, without the syntactic sugar:

```
import commonmark
parser = commonmark.Parser()
ast = parser.parse("Hello *World*")

renderer = commonmark.HtmlRenderer()
html = renderer.render(ast)
print(html) # <p>Hello <em>World</em></p>
```

(continues on next page)

(continued from previous page)

```
# inspecting the abstract syntax tree
json = commonmark.dumpJSON(ast)
commonmark.dumpAST(ast) # pretty print generated AST structure
```

There is also a CLI:

```
$ cmark README.md -o README.html
$ cmark README.md -o README.json -aj # output AST as JSON
$ cmark README.md -a # pretty print generated AST structure
$ cmark -h
usage: cmark [-h] [-o [O]] [-a] [-aj] [infile]

Process Markdown according to the CommonMark specification.

positional arguments:
  infile          Input Markdown file to parse, defaults to stdin

optional arguments:
  -h, --help      show this help message and exit
  -o [O]          Output HTML/JSON file, defaults to stdout
  -a              Print formatted AST
  -aj             Output JSON AST
```

1.3 Contributing

If you would like to offer suggestions/optimizations/bugfixes through pull requests please do! Also if you find an error in the parser/renderer that isn't caught by the current test suite please open a new issue and I would also suggest you send the [commonmark.js](#) project a pull request adding your test to the existing test suite.

1.4 Tests

To work on CommonMark-py, you will need to be able to run the test suite to make sure your changes don't break anything. To run the tests, you can do something like this:

```
$ pyvenv venv
$ ./venv/bin/python setup.py develop test
```

The tests script, `commonmark/tests/run_spec_tests.py`, is pretty much a devtool. As well as running all the tests embedded in `spec.txt` it also allows you to run specific tests using the `-t` argument, provide information about passed tests with `-p`, percentage passed by category of test with `-s`, and enter markdown interactively with `-i` (In interactive mode end a block by inputting a line with just `end`, to quit do the same but with `quit`). `-d` can be used to print call tracing.

```
$ ./venv/bin/python commonmark/tests/run_spec_tests.py -h
usage: run_spec_tests.py [-h] [-t T] [-p] [-f] [-i] [-d] [-np] [-s]

script to run the CommonMark specification tests against the CommonMark-py
parser.

optional arguments:
  -h, --help      show this help message and exit
```

(continues on next page)

(continued from previous page)

-t T	Single test to run or comma separated list of tests (-t 10 or -t 10,11, →12,13)
-p	Print passed test information
-f	Print failed tests (during -np...)
-i	Interactive Markdown input mode
-d	Debug, trace calls
-np	Only print section header, tick, or cross
-s	Print percent of tests passed by category

1.5 Authors

- Bibek Kafle
- Roland Shoemaker
- Nikolas Nyby

1.5.1 API

HTML

reStructuredText

Parser

Node