
em

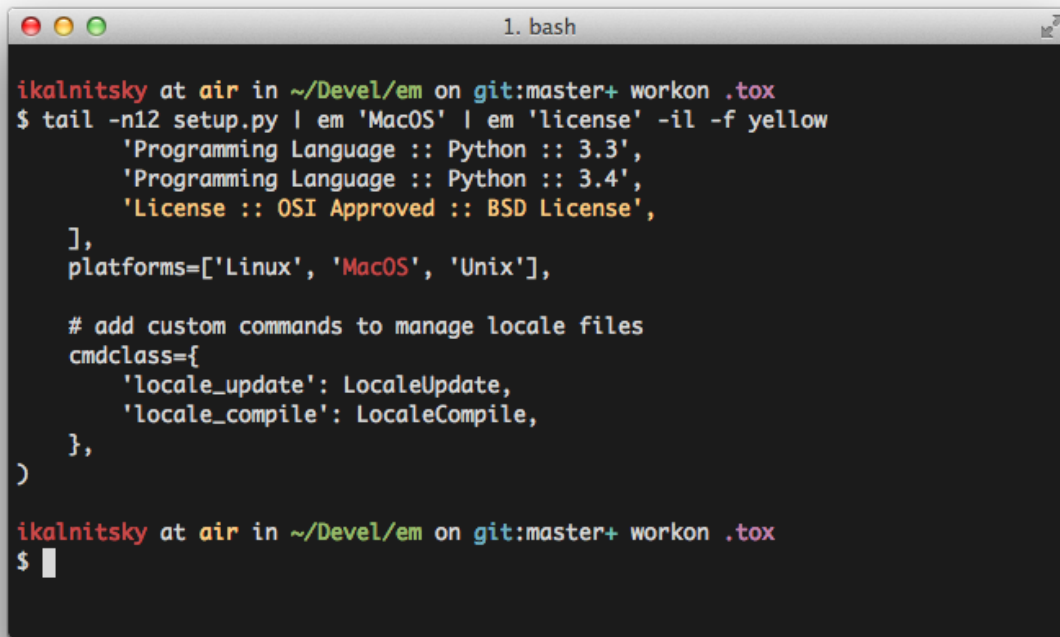
Release 0.4.0

Sep 27, 2017

Contents

1	Installation	3
2	Quickstart	5
3	Options	7
4	Contribute	9

Em is a terminal tool that prints FILE(s), or standard input to standard output and highlights the expressions that are matched the PATTERN.

A screenshot of a terminal window titled "1. bash". The prompt is "ikalnitsky at air in ~/Devel/em on git:master+ workon .tox". The command executed is "\$ tail -n12 setup.py | em 'MacOS' | em 'license' -il -f yellow". The output shows several lines of text from the setup.py file, with some lines highlighted in yellow. The highlighted lines are: "'Programming Language :: Python :: 3.3'", "'Programming Language :: Python :: 3.4'", and "'License :: OSI Approved :: BSD License'". The output ends with a closing parenthesis ")", and the prompt returns to "\$".

```
ikalnitsky at air in ~/Devel/em on git:master+ workon .tox
$ tail -n12 setup.py | em 'MacOS' | em 'license' -il -f yellow
    'Programming Language :: Python :: 3.3',
    'Programming Language :: Python :: 3.4',
    'License :: OSI Approved :: BSD License',
],
platforms=['Linux', 'MacOS', 'Unix'],

# add custom commands to manage locale files
cmdclass={
    'locale_update': LocaleUpdate,
    'locale_compile': LocaleCompile,
},
)

ikalnitsky at air in ~/Devel/em on git:master+ workon .tox
$
```

The expression will be highlighted iff the terminal is ANSI-compatible. This criterion is met by most popular terminal emulators on Linux, Unix and MacOS.

CHAPTER 1

Installation

Em is a small script written in Python without any further dependencies but things that are shipped with Python 2.7, Python 3.2 or above. Quick installation:

```
$ [sudo] pip install em
```

Note: For some reason the Python 2.6 support has been added, but `argparse` is required and will be installed from PyPI.

CHAPTER 2

Quickstart

Em is cool and easy to use. The most practical usecase for me:

```
$ tail -f /path/to/log | em "ERROR" | em "iphone:" -f green
```

The example above highlights all occurrences of «ERROR» in red and all occurrences of «iphone:» in green when watching the log.

But you can use em in the following way:

```
$ em "ERROR|CRITICAL" /path/to/log
```

which prints the log and highlights all occurrences of «ERROR» or «CRITICAL» in red.

CHAPTER 3

Options

Here is a command line interface of Em:

```
usage: em [OPTION]... PATTERN [FILE]...
```

Em **is** a terminal tool that prints FILE(s), **or** standard **input** to standard output **and** highlights the expressions that are matched the PATTERN.

positional arguments:

PATTERN	a pattern to highlight
FILE	search for pattern in these file(s)

optional arguments:

-f FORMAT, --format FORMAT	a color to highlight matched expressions
-i, --ignore-case	ignore case distinctions
-l, --line-mode	highlight entire line
-v, --version	show program's version number and exit
-h, --help	show this help message and exit

With no FILE, **or** when FILE **is** -, read standard **input**. The FORMAT option may be one of: BOLD, UNDERLINE, [ON]GREY, [ON]RED, [ON]GREEN, [ON]YELLOW, [ON]BLUE, [ON]MAGENTA, [ON]CYAN **or** [ON]WHITE. Default **is** RED.

The CLI is clear, but some option descriptions are below:

--format	Highlight matched PATTERN with that color. (default: RED)
--ignore-case	Case insensitive search for the PATTERN.
--line-mode	Highlights the entire line if PATTERN was found in the line.

CHAPTER 4

Contribute

Found a bug? Have a good idea for improving Em? Go to [em's github](#) page and create a new issue or fork. Also, if you like what I'm doing I would appreciate some support through [gittip](#).