My Checker Documentation

Release 0.1.1

Michael Reuter

May 26, 2016

Contents

1 Features **3**

1.1 Contents: 3

1.2 Feedback 6

Python Module Index **7**

Quick math checker

CHAPTER 1

Features

- Simple expression evaluation.

1.1 Contents:

1.1.1 Installation

At the command line either via easy_install or pip:

```
$ easy_install my_checker
$ pip install my_checker
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv my_checker
$ pip install my_checker
```

1.1.2 Usage

To use My Checker in a project:

```
>>> import my_checker
>>> if my_checker.check_math(6 * 9 - 12, 42):
...     print("The answer!")
... else:
...     print("Sorry for the inconvenience.")
The answer!
```

1.1.3 Contributing

Contributions are welcome, but this is a tiny package that I'm using for testing, so it won't be very useful to anyone but me.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/mareuter/my_checker/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.

Fix Bugs

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

Implement Features

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

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/mareuter/my_checker/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 :)

Get Started!

Ready to contribute? Here's how to set up *my_checker* for local development.

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

```
    $ git clone git@github.com:your_name_here/my_checker.git
```

3. Create a branch for local development:

```
    $ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you're done making changes, check that your changes pass style and unit tests, including testing other Python versions with tox:

```
    $ tox
```

To get tox, just pip install it.

5. 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
```

6. Submit a pull request through the GitHub website.

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, and 3.5, and for PyPy. Check https://travis-ci.org/mareuter/my_checker under pull requests for active pull requests or run the `tox` command and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ py.test test/test_my_checker.py
```

1.1.4 Credits

Development Lead

- Michael Reuter <mareuternh@gmail.com>

Contributors

None yet. Why not be the first?

1.1.5 History

0.1.1 (2016-05-25)

- Added doctest as usage example.

0.1.0 (2016-05-23)

- First release on PyPI.

1.1.6 my_checker

my_checker package

Submodules

my_checker.my_checker module

`my_checker.my_checker.`**`check_math`**(*expression*, *answer*)

Check a math expression with an answer.

Parameters

- **`expression`** (*`math equation`*) – The math expression to check.
- **`answer`** (*`numeric`*) – The answer to the expression.

Returns True if answer matches evaluated equation, False if not.

Return type bool

Module contents

1.2 Feedback

If you have any suggestions or questions about **My Checker** feel free to email me at mareuternh@gmail.com.

If you encounter any errors or problems with **My Checker**, please let me know! Open an Issue at the GitHub http://github.com/mareuter/my_checker main repository.

Python Module Index

m

my_checker, 6

my_checker.my_checker, 6

Index

C

check_math() (in module my_checker.my_checker), 6

M

my_checker (module), 6
my_checker.my_checker (module), 6