amargan Documentation

Release 0.0.1

Francis Horsman

Oct 08, 2017

Contents

```
>>> print('Anagrams for everyone')
```

CHAPTER 1

ALPHA

About Amargan

```
>>> print('Anagrams for everyone')
```

Reasons to use amargan

- A simple but powerful pythonic interface

```
>>> from amargan import Amargan
...
... with open('words.txt') as iterator:
...     anagrams = Amargan(iterator)
... anagrams['hello']
set(['elloh' 'hello' 'lehol'])
```

- A powerful command-line tool

```
$ find_anagrams -i words.txt hello
elloh hello lehol

$ amargan -i words.txt hello
elloh hello lehol
```

- Extensive configuration options
  - Case (in)sensitivity
  - Exclusion of word from results
  - Output formatting (one per line, multiple-per-line, custom seperator)
  - Output to a file
  - Read from a file

  - Use existing dictionary of words
- Time complexity: O(1)
- Space complexity: O(n) where n = number of words in dictionary.
- Memory efficient, uses iterators extensively.
- Add and remove word(s) from the dictionary.
- Extensively tested with excellent code and branch coverage.
- Extensive error checking with a rich set of checked exceptions.
- Uses Certifiable if available to catch runtime type and parameter validation errors.
- JSON serializable and reconstitutable.
- Fully documented
- Free software: MIT license
- Documentation: https://amargan.readthedocs.io

amargan-pyp

amargan package

Submodules

amargan.amargan module

amargan.cli module

Console script for amargan.

amargan.errors module

amargan.iterators module

amargan.utils module

Module contents

Top-level package for amargan.

tests package

Module contents

Unit test package for amargan.

Installation

Stable release

To install amargan, run this command in your terminal:

```
$ pip install amargan
```

This is the preferred method to install amargan, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

From sources

The sources for amargan can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/sys-git/amargan
```

Or download the tarball:

```
$ curl  -OL https://github.com/sys-git/amargan/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

Usage

Programmatic

Amargan takes any iterable that yields strings, thus making it memory efficient:

Example: From the contents of a file

```
>>> from amargan import Amargan
...
... with open(filename) as fp:
...     anagrams = Amargan(fp.readlines())
... anagrams['abc']
set(['abc', 'acb', 'cba'])
... anagrams.for_word('abc')
set(['abc', 'acb', 'cba'])
```

Example: From an open file

```
>>> from amargan import Amargan
...
... with open(filename) as fp:
...     anagrams = Amargan(fp)
... anagrams['abc']
set(['abc', 'acb', 'cba'])
```

```
... anagrams.for_word('abc')
set(['abc', 'acb', 'cba'])
```

Example: From a StringIO

```
>>> from amargan import Amargan
...
... sio = six.StringIO(buf='cba\nabc\nacb\n')
... anagrams = Amargan(sio)
... anagrams['abc']
set(['abc', 'acb', 'cba'])
... anagrams.for_word('abc')
set(['abc', 'acb', 'cba'])
```

Example: From a list

```
>>> from amargan import Amargan
...
... anagrams = Amargan(['cba', 'abc', 'acb'])
... anagrams['abc']
set(['abc', 'acb', 'cba'])
... anagrams.for_word('abc')
set(['abc', 'acb', 'cba'])
```

There are configurable *Iterators* to allow you to read from a file using a non-default configuration.

For example, to iterate over a multi-line file containing words separated by a comma:

```
>>> from amargan import Amargan, Iterator, IteratorType
...
... with open(filename) as fp:
...     iterator = Iterator(IteratorType.multi_per_line, sep=',')
...     anagrams = Amargan(iterator(fp))
... anagrams['abc']
set(['abc', 'acb', 'cba'])
```

To iterate over a multi-line file containing lines of one or more words separated by a whitespace (the default iterator configuration):

```
>>> from amargan import Amargan
...
... with open(filename) as iterator:
...     anagrams = Amargan(iterator)
... anagrams['abc']
set(['abc', 'acb', 'cba'])
```

Add and remove words from the dictionary:

```
>>> from amargan import Amargan
...
... anagrams = Amargan()
... anagrams['acb']
frozenset()
... anagrams += 'abc acb cab'
... anagrams['acb']
set(['abc', 'acb', 'cba'])
... anagrams -= 'acb'
```

```
... anagrams['acb']
set(['abc', 'cba'])
```

```
>>> from amargan import Amargan
...
... anagrams = Amargan()
... anagrams['acb']
frozenset()
... x = anagrams + 'abc acb cab'
... x
Amargan(True - 1)
... x['acb']
set(['abc', 'acb', 'cba'])
... x = anagrams - 'acb'
... x['acb']
set(['abc', 'cba'])
```

Command-line

To use amargan from the command-line for single words (anagrams are ordered and contain the original word by default):

```
$ find_anagrams -i words.txt hello
elloh hello lehol
```

and for multiple words:

```
$ find_anagrams --ip=words.txt hello world
elloh hello lehol
lordw rlwdo world
```

and with options:

```
$ find_anagrams --exclude --output-iterator=one_per_line --case-sensitive --ip=words.
↪txt Hello
elloH
leHol
```

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:

Types of Contributions

Report Bugs

Report bugs at https://github.com/sys-git/amargan/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" and "help wanted" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

Write Documentation

amargan could always use more documentation, whether as part of the official amargan 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/sys-git/amargan/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 *amargan* for local development.

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

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

3. Create a virtualenv with all dependencies:

```
$ make build
```

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:

```
$ make flake8
$ make 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.

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, 3.5, 3.6 and for PyPy. Check https://travis-ci.org/sys-git/amargan/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_amargan
```

Credits

Development Lead

- Francis Horsman <francis.horsman@gmail.com>

Contributors

None yet. Why not be the first?

History

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

t

Index