
WordCount Documentation

Release 0.1

Harsha

Jun 05, 2019

Contents

1	Purpose	1
1.1	Zipf's law	1
2	Dependencies	3
2.1	Required	3
2.2	Optional	3
3	Usage	5
3.1	How to clone the code	5
3.2	Make	5
3.3	Snakemake	5
3.4	Where to find the results	5
4	Credit and inspiration	7
5	Exercises (documentation lesson)	9
5.1	Before you start	9
5.2	Basic	9
5.3	Advanced	10
5.4	Meta	10

CHAPTER 1

Purpose

The purpose of this exercise is to test readthedocs webhooks and show how easy it is to make an efficient documentation.

1.1 Zipf's law

This is some weird word count stuff that I don't really get at this point.

2.1 Required

- Python
- Numpy
- Matplotlib
- Make or Snakemake

2.2 Optional

- Docker

3.1 How to clone the code

Write me ...

3.2 Make

Generate all results:

```
$ make -f Makefile_all
```

3.3 Snakemake

Write me ...

3.4 Where to find the results

Write me ...

CHAPTER 4

Credit and inspiration

Inspired by and derived from <https://hpc-carpentry.github.io/hpc-python/> which is distributed under CC-BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>).

Exercises (documentation lesson)

5.1 Before you start

- Discuss the exercise idea with the classroom.
- Distribute exercises among groups of 2-3 persons.
- Open a GitHub issue and inform the community about the problem and how you plan to solve it. Discuss why we do this.
- Fork this project.
- Commit to your fork. In your commit message auto-close the issue you have addressed.
- Submit a pull request.
- We then review the pull requests.
- After the pull requests are merged we verify that documentation updates itself.

5.2 Basic

- Document the purpose of this example code.
- Document how to clone the code.
- Describe the project tree structure.
- Write a sentence or two about Zipf's law and link to Wikipedia (coordinate with the group working on the previous exercise).
- Document how to check the code style with `pycodestyle`.
- Give other developers hints on how they can contribute to the documentation.
- Document how to build the documentation locally (coordinate with the group working on the previous exercise).
- Add an example output.

- Add an example plot (coordinate with the group working on the previous exercise).
- Document where/how to ask for help.
- Add a math equation somewhere.

5.3 Advanced

- Add a test and document how to run it.
- Add the possibility to auto-document Python code.

5.4 Meta

- Add new exercises ideas for future workshops (edit this file).