

---

# Flashcards

*Release 0.1.0*

Nov 29, 2018



---

## Contents

---

<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	What flashcards are . . . . .	1
1.3	How does this work . . . . .	2
1.4	Usage . . . . .	2
1.5	Documentation . . . . .	2
1.6	Development . . . . .	2
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Reference</b>	<b>9</b>
4.1	flashcards . . . . .	9
<b>5</b>	<b>Contributing</b>	<b>11</b>
5.1	Bug reports . . . . .	11
5.2	Documentation improvements . . . . .	11
5.3	Feature requests and feedback . . . . .	11
5.4	Development . . . . .	12
<b>6</b>	<b>Authors</b>	<b>13</b>
<b>7</b>	<b>Changelog</b>	<b>15</b>
7.1	0.3.0 (2018-11-27) . . . . .	15
7.2	0.1.0 (2017-03-31) . . . . .	15
<b>8</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



# CHAPTER 1

---

## Overview

---

docs	
tests	
package	

small cli tool to study using fl

- Free software: BSD license

## 1.1 Installation

```
pip3 install python-flashcards
```

## 1.2 What flashcards are

A flashcard or flash card is a set of cards bearing information, as words or numbers, on either or both sides, used in classroom drills or in private study. One writes a question on a card and an answer overleaf.  
[\[Wikipedia\]](#)

## 1.3 How does this work

`python-flashcards` is a small tool, which receives cards from a YAML file, and shows them in a random order so you can practice.

YAML format:

```
-
  topic: The topic I will say out loud
  content: The information I'll check after saying out loud what I know
  keywords: reference, words
-
  topic: Python
  content: Is a widely used high-level programming language for general-purpose_
  ↪programming,
    created by Guido van Rossum and first released in 1991.
  keywords: programming, language
```

Being keyword the only optional.

## 1.4 Usage

Let's suppose `anatomy.yaml` is your file with information related to anatomy.

```
flashcards anatomy.yaml
```

If you want the cards in order:

```
flashcards --ordered meds.yaml
```

If you want to show the hide the topic instead of the content:

```
flashcards --inverted meds.yaml
```

For more help:

```
flashcards -h
```

## 1.5 Documentation

<https://python-flashcards.readthedocs.io/>

## 1.6 Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	<pre>set PYTEST_ADDOPTS=--cov-append tox</pre>
Other	<pre>PYTEST_ADDOPTS=--cov-append tox</pre>





## CHAPTER 2

---

### Installation

---

At the command line:

```
pip install flashcards
```



## CHAPTER 3

---

### Usage

---

To use Flashcards in a project:

```
import flashcards
```



## CHAPTER 4

---

Reference

---

### 4.1 flashcards



Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

### 5.1 Bug reports

When [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.

### 5.2 Documentation improvements

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

### 5.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/woile/flashcards/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 code contributions are welcome :)

## 5.4 Development

To set up *flashcards* for local development:

1. Fork *flashcards* (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/flashcards.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, run all the checks, doc builder and spell checker with *tox* one command:

```
tox
```

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.

### 5.4.1 Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run *tox*)<sup>1</sup>.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to *CHANGELOG.rst* about the changes.
4. Add yourself to *AUTHORS.rst*.

### 5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

---

<sup>1</sup> If you don’t have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.  
It will be slower though ...



## CHAPTER 6

---

### Authors

---

- Santiago Fraire Willemoes - <https://woile.github.io>
- Elliott King



#### **7.1 0.3.0 (2018-11-27)**

- Add ability to hide topic instead of content.

#### **7.2 0.1.0 (2017-03-31)**

- First release on PyPI.



## CHAPTER 8

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**f**

flashcards, 9





## F

flashcards (module), [9](#)