

Flashcards

Leon Rische

[2019-07-22 Mon 22:57]

Contents

1	Cards	1
2	Compound Cards / Deletions	2
3	Hotkeys	OUTDATED 3
4	Text Formatting	4
5	Export / Import	OUTDATED 4
6	Import from Anki	4
7	Misc	5

I'm building and using a space repetition system (like Anki). It started as a web application but I'm merging more and more of it into Emacs.

For the documentation of the emacs version, see Emacs Flashcards.

1 Cards

A card has a front and a back side, usually with a question and its answer. Both can be styled using **markdown**.

Some types of cards can have a plain-text answer, too.

Each card an ease E value (default 2.5) assigned to it that determines how often it needs to be repeated.

The first few intervals (one minute, ten minutes, one day) are pretty short and meant for **learning** the card, after that the next **review** intervals I_{n+1} depend on the cards ease.

$$I_{n+1} = E \cdot I_n$$

1.1 Type: Front / Back

Equivalent to writing stuff on a 3x5 card, useful to learn translations and definitions of things.

1.2 Type: Question / Answer

Has a third field, 'Answer'. When reviewing it, you'll need to type it in as text that is compared to the stored answer. The "Back / Explanation" field is optional and can be used to write down some more details about the answer.

After submitting, you can compare your answer with the stored one. If they are the same, the "Again" button is hidden, but you can rate how easy it was to recall the answer.

Otherwise the bottom of the card shows a visualisation of the difference between the two answers and you can determine yourself if was minor (**s** instead of **str**) or if you need to repeat that card.

2 Compound Cards / Deletions

A way to create and edit multiple cards that belong together at the same time, e.g. for Cloze Tests.

Cards must have at least one deletion, surrounded by `{{` and `}}` in the card front field.

Let's say you want to learn the first few digits of π . A deletion might look like this:

3.{{1415}}{{9265}}

2.1 Type: Cloze Deletion

Removes just one of the marked deletions at a time, creating a "hole" in the card.

2.2 Type: Enumeration

Removes one of the marked deletions and all the ones after it, useful for learning lists of things.

2.3 Caution!

Right now editing deletions in a way that keeps the overall count the same seems to work, adding or removing them might break stuff.

3 Hotkeys

OUTDATED

3.1 Everywhere

- `p h =>` Open Page "Home"
- `p r =>` Open Page "Review"
- `p d =>` Open Page "Decks"
- `p s =>` Open Page "Settings"
- `p ? =>` Open Page "Search"

3.2 Deck List / Card List

Each deck / card is marked with a unique combination of one or more letters, use `c $combination` to open one of them.

3.3 Deck List

- `a =>` Create a new deck

3.4 Card List

- `a c =>` Add a new card
- `a d =>` Add a new deletion
- `a t =>` Add a new translation

3.5 Review

- `enter =>` Flip a card
- `a =>` Rate as "Again"
- `h =>` Rate as "Hard"
- `g =>` Rate as "Good"
- `e =>` Rate as "Easy"

4 Text Formatting

Cards fronts and back are formatted using org-mode markup, for better integration with Emacs.

5 Export / Import

OUTDATED

The import feature (Deck -> Import) uses a pretty simple JSON based format:

```
language=json,label=,caption=,captionpos=b,numbers=none "cards":
["<card>"], "generator_cards": ["<generator_card>"]
```

Both **cards** and **cloze_deletions** are optional and can be empty arrays, too.

Cards need to look like this:

```
language=json,label=,caption=,captionpos=b,numbers=none "front":
"front ...", "back": "back ...", "answer": "answer ...", "type": "<card_type>
", "suspended": true|false,
```

answer, **back** and **suspended** are optional. Valid types are:

1. "simple" for Front / Back cards
2. "text_input" for Question / Answer cards
3. "keys" for Keystroke cards

Generator cards need to look like this:

```
language=json,label=,caption=,captionpos=b,numbers=none "front":
"front ...", "type": "<card_type> "
```

Valid types are:

1. "cloze" for Cloze Deletions (holes in the text)
2. "enum" for Enumerations

6 Import from Anki

Anki import has been deprecated in favor of an internal JSON based format, so it is necessary to convert between the two formats first.

1. Inside Anki, select a deck and export it as **.txt**

2. Use the ruby script below (our your own) to convert it to JSON
3. On the flashcards site, select a deck, "Import" and upload the generated file

```

language=Ruby,label= ,caption= ,captionpos=b,numbers=none require
"json"
if ARGV.size != 2 puts "USAGE: convert input.txt output.json" else
cards = []
File.open(ARGV[0]).each do |line| tokens = line.chomp.split("\t") cards «
front: tokens[0], back: tokens[1], type: "simple" end
File.open(ARGV[1], "w") do |file| payload = cards: cards file.puts(payload.to_json)endend

```

7 Misc

7.1 Time Format (in card headers)

- Y, years
- M, months
- D, days
- h, hours
- m, minutes
- s, seconds