Introduction to Python. Hometask

You are proposed to implement Python RSS-reader using **python 3.8**.

The task consists of few iterations. Do not start new iteration if the previous one is not implemented yet.

Common requirements

- It is mandatory to use `argparse` module.
- Codebase must be covered with unit tests with at least 50% coverage.
- In case of any mistakes utility should print human-readable error explanation. Exception tracebacks in stdout are prohibited in final version of application.
- Docstrings are mandatory for all methods, classes, functions and modules.
- Code must correspond to `pep8` (use `pycodestyle` utility for self-check).
    - You can set line length up to 120 symbols.
- Commit messages should provide correct and helpful information about changes in commit. Messages like `Fix bug`, `Tried to make workable`, `Temp commit` and `Finally works` are prohibited.
- Usage of external APIs is prohibited (except of APIs for receiving RSS)

[Iteration 1] One-shot command-line RSS reader.

RSS reader should be a command-line utility which receives RSS URL and prints results in human-readable format.

You are free to choose format of the news console output. The textbox below provides an example of how it can be implemented:

```
$ rss_reader.py "https://news.yahoo.com/rss/" --limit 1

Feed: Yahoo News - Latest News & Headlines

Title: Nestor heads into Georgia after tornados damage Florida
Date: Sun, 20 Oct 2019 04:21:44 +0300
Link: https://news.yahoo.com/wet-weekend-tropical-storm-warnings-131131925.html

[image 2: Nestor heads into Georgia after tornados damage Florida][2]Nestor raced across
homes and a school in central Florida while sparing areas of the Florida Panhandle devas
off Florida's northern Gulf Coast in a lightly populated area of the state, the National
march across a swath of the U.S. Southeast.


Links:
[1]: https://news.yahoo.com/wet-weekend-tropical-storm-warnings-131131925.html (link)
[2]: http://l2.yimg.com/uu/api/res/1.2/Liyq2kH4HqlYHaS5BmZWpw--/YXBwaWQ9eXRhY2h5b247aD04
```

Utility should provide the following interface:

```
usage: rss_reader.py [-h] [--version] [--json] [--verbose] [--limit LIMIT]
                     source

Pure Python command-line RSS reader.

positional arguments:
  source         RSS URL

optional arguments:
  -h, --help     show this help message and exit
  --version      Print version info
  --json         Print result as JSON in stdout
  --verbose      Outputs verbose status messages
  --limit LIMIT  Limit news topics if this parameter provided
```

In case of using `--json` argument your utility should convert the news into JSON format. You should come up with the JSON structure on you own and describe it in the README.md file for your repository or in a separate documentation file.

The `--limit` argument should also affect JSON generation.

With the argument `--verbose` your program should print all logs in stdout.

Withe the argument `--version` your program should print in stdout it's current version and complete it's work. The version supposed to change with every iteration.

[Iteration 2] Distribution

- Utility should be wrapped into distribution package with `setuptools`.
- This package should export CLI utility named `rss-reader`.

> Note: Double-check, that your utility works correctly after its new package was installed on a clean machine.

[Iteration 3] News caching

The RSS news should be stored in a local storage while reading. The way and format of this storage you can choose yourself. Please describe it in a separate section of README.md or in the documentation.

New optional argument `--date` must be added to your utility. It should take a date ir `%Y%m%d` format. For example: `--date 20191020`

The cashed news can be read with it. The new from the specified day will be printed out. If the news are not found return an error.

If the `--date` argument is not provided, the utility should work like in the previous iterations.

[Iteration 4] Format converter

You should implement the conversion of news in at least two of the suggested format `.mobi`, `.epub`, `.fb2`, `.html`, `.pdf`

New optional argument must be added to your utility. This argument receives the path where new file will be saved. The arguments should represents which format will be generated.

For example: `--to-mobi` or `--to-fb2` or `--to-epub`

You can choose yourself the way in which the news will be displayed, but the final text result should contain pictures and links, if they exist in the original article and if the format permits to store this type of data.

* [Iteration 5] Output colorization

> Note: An optional iteration, it is not necessary to implement it. You can move on with it only if all the previous iterations (from 1 to 4) are completely implemented.

You should add new optional argument `--colorize`, that will print the result of the utility in colorized mode.

If the argument is not provided, the utility should work like in the previous iterations.

> Note: Take a look at the colorize library

* [Iteration 6] Web-server

> Note: An optional iteration, it is not necessary to implement it. You can move on with it only if all the previous iterations (from 1 to 4) are completely implemented. Introduction to Python course does not cover the topics that are needed for the implementation of this part.

There are several mandatory requirements in this iteration:
* `Docker` + `docker-compose` usage (at least 2 containers: one for web-application, one for DB)
* Web application should provide all the implemented in the previous parts of the task functionality, using the REST API:
- One-shot conversion from RSS to Human readable format
- Server-side news caching
- Conversion in epub, mobi, fb2 or other formats

Feel free to choose the way of implementation, libraries and frameworks. (We suggest you `Django Rest Framework` + `PostgreSQL` combination)

You can implement any functionality that you want. The only requirement is to add the description into README file or update project documentation, for example:
* authorization/authentication
* automatic scheduled news update

* adding new RSS sources using API

Implementations will be checked with the latest cPython interpreter of 3.8 branch.

> Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live. Code for readability. **John F. Woods**