1. **Using threading library to verify URL's.**

В файле links.txt содержится список url-адресов которые нужно провалидироавть на доступность.

Нам необходимо наиболее эффективное решение. Используйте библиотеку threading, чтобы инициировать множество web-запросов одновременно.

Результат выполнения скрипта сохраните в файл в формате JSON (см. пример ниже)

Примечание:

* Используйте библиотеку requests (pip install requests)
* Проперти Response.ok может быть полезной для определения валидности респонза (github/requests)

Пример результата:

[

{

"url": "https://www.does\_not\_exist.com/",

"is\_ok": false,

"status\_code": null

},

{

"url": "http://lucumr.pocoo.org/",

"is\_ok": true,

"status\_code": 200

},

{

"url": "http://jinja.pocoo.org/docs/",

"is\_ok": true,

"status\_code": 200

}

]

1. **Multiprocessing library.**

Для решения задачи #5 (Multiples of 3 and 5) было реализовано множество различных алгоритмов. В файле multiples\_of\_three\_and\_five.py перечислены наиболее популярные из них.

Давайте выясним какой из предложенных алгоритмов является самым эффективным.

Используйте библиотеку multiprocessing, чтобы запустить параллельное выполнение всех 4-х функций. Вам также понадобится контекстный менеджер timer

Ответьте на вопрос:

* почему мы не используем библиотеку threading для решения этой задачи?

Check yourself:

bash$ python multiples\_of\_three\_and\_five.py

Output:

math\_formula() -> 0.0 sec

several\_for\_loops() -> 28.89 sec

iterate\_over\_fifteen() -> 41.32 sec

simple\_iteration() -> 66.87 sec