

# Основы работы с потоками на Python

## Основы работы с потоками и объектами Concurrency.

### Лабораторная работа № 3. Задачи.

#### Комплект 1: Библиотека Asyncio.

- 1.1: Создайте простую асинхронную функцию, которая в бесконечном цикле отображает текущее время с интервалом 1 секунда, и запустите ее с помощью `asyncio`. Завершать программу можно через комбинацию клавиш типа `Ctrl + C` или `Ctrl + Break`.
- 1.2: Улучшите предыдущую программу. Распечатывайте текущие дату и время в одной и той же строке (используйте символ возврата каретки `'\r'`) разными цветами с помощью библиотеки `termcolor`. Добавьте обработку события нажатия клавиши `Esc` с помощью библиотеки `rpymp`, чтобы можно было выйти из программы.
- 1.3: Разработайте программу, которая использует `asyncio.gather` для выполнения двух асинхронных задач параллельно и затем обрабатывает результаты в отдельной функции в цикле.
- 1.4: Улучшите предыдущую программу, сделав параллельно запрос к публичному веб-серверу и к любой публичной базе данных с распечаткой результатов этих вызовов. Например сделайте запросы к публичному ресурсу последовательностей РНК - RNACentral: <https://rnacentral.org/about-us>.

```
# https://rnacentral.org/api
# URL вебсервера-
WEB_SERVER_URL = "https://rnacentral.org/api/v1/rna/"

# https://rnacentral.org/help/public-database
# Строкаподключениякбазеданных
DB_CONNECTION_STRING = \
"postgres://reader:NWDMCE5xdipJRp@hh-pgsql-public.ebi.ac.uk:5432/pfmegrnargs"
```

В этом случае вам потребуются библиотеки `aiohttp` - для любых HTTP запросов и `asyncpg` - для базы данных Postgres ресурса RNACentral. Отформатируйте красиво JSON сообщение с помощью модуля `json`, если это URL типа `https://rnacentral.org/api/v1/rna/`.

- 1.5: Создайте асинхронный веб-скрапер, используя `aiohttp` и `asyncio`, который собирает информацию из нескольких веб-страниц одновременно. Вынесите код скрапера в отдельный класс. Загружайте список URLs из отдельного файла.
- 1.6: Улучшите асинхронный веб-скрапер из предыдущей задачи. Преобразуйте его в класс асинхронного менеджера контекста и используйте вместе с `async with` изнутри асинхронной функции `main`, которая вызывается через `asyncio.run(main())`.
- 1.7: Реализуйте сервер и клиент с использованием библиотек `asyncio` и HTTPS протокола для отправки текстовых JSON сообщений, позволяющий передавать текстовые сообщения в обе стороны в цикле с клавиатуры в режиме `echo`.