

# Лабораторная работа

## Цели

1. Понять, как ведут себя io bound операции в потоках и процессах в python
2. Понять, как ведут себя cpu bound операции в потоках и процессах в python
3. Сравнить время выполнения реализованных программ

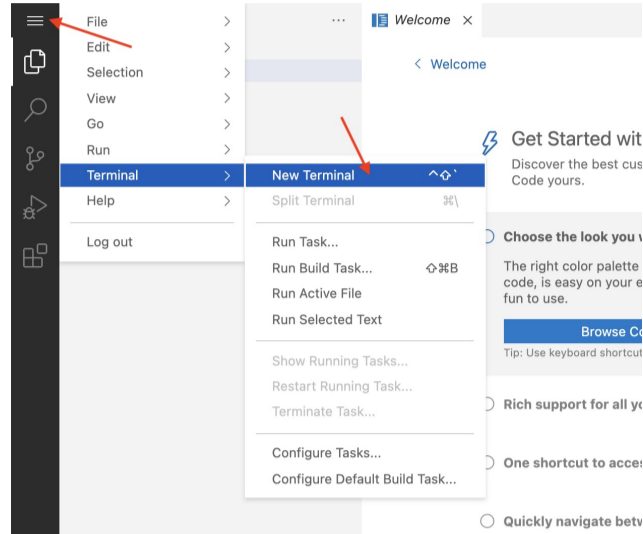
## Задания

Шаблон отчета — [https://docs.google.com/document/d/10hgfmPYGmbyklBHSZ90gQBej-w\\_Qwj4F/edit](https://docs.google.com/document/d/10hgfmPYGmbyklBHSZ90gQBej-w_Qwj4F/edit)

В задании нужно дополнить файлы:

```
io_ops/in_proc.py
io_ops/in_threads.py
io_ops/seq.py
cpu_ops/in_proc.py
cpu_ops/in_threads.py
cpu_ops/seq.py
```

Для запуска нужно открыть терминал:



В терминале запустить скрипт. Например:

```
python3 io_ops/in_proc.py
```

Все необходимые зависимости для выполнения задания уже установлены, их можно посмотреть в файле requirements.txt

## IO bound операции

### Задание 1. Последовательные запросы.

Файл **io\_ops/seq.py**

Сделать 10 синхронных запросов на сайт <https://api.covidtracking.com/v1/us/current.json>. Для запросов использовать библиотеку [requests](#). Время выполнения приложите к отчету.

### Задание 2. Запросы в тредах

Файл **io\_ops/in\_threads.py**

Сделать 10 запросов на сайт <https://api.covidtracking.com/v1/us/current.json> в разных тредах. Для запросов использовать библиотеку [requests](#). Время выполнения приложите к отчету.

### Задание 3. Запросы в процессах

Файл **io\_ops/in\_proc.py**

Сделать 10 запросов на сайт <https://api.covidtracking.com/v1/us/current.json> в разных процессах. Для запросов использовать библиотеку [requests](#). Время выполнения приложите к отчету.

## Выводы из заданий 1-3

Напишите выводы после выполнения заданий. В выводе нужно:

- ответить, выполняются ли параллельно io bound операции в потоках/процессах
- обосновать, почему они выполняются или не выполняются параллельно

## CPU bound операции

Функция, которую будем использовать в заданиях, связанных с cpu bound операциями:

```
def countdown():
    i = 0
    begin = time.time()
    while i < 5_000_000:
        i += 1
    print(f"duration: {time.time() - begin}")
```

### Задание 4. Синхронное выполнение

Файл **cpu\_ops/seq.py**

10 раз синхронно вызвать функцию **countdown()**. Время выполнения и код программы приложите к отчету.

### Задание 5. Выполнение в тредах

Файл **cpu\_ops/in\_threads.py**

Вызвать функцию **countdown()** в 10 разных тредах. Время выполнения приложите к отчету.

### Задание 6. Выполнение в процессах

Файл **cpu\_ops/in\_proc.py**

Вызвать функцию **countdown()** в 10 разных процессах. Время выполнения приложите к отчету.

## Выводы из заданий 4-6

Напишите выводы после выполнения заданий. В выводе нужно:

- ответить, выполняются ли параллельно cpu bound операции в потоках/процессах
- обосновать, почему они выполняются или не выполняются параллельно