**Multiprocessing, Multithreading, Асинхронность, GIL**

***Multiprocessing (многопроцессорное выполнение)*** – заключается в запуске отдельных **процессов**. При этом, каждый процесс имеет свой **собственный интерпретатор Python и отдельное пространство памяти**. Это означает, что каждый процесс может выполняться независимо от других.

***Multithreading (многопоточное выполнение)*** - использует **потоки** внутри **одного процесса**. Потоки разделяют общее пространство памяти процесса и могут общаться между собой.

***GIL (Global Interpreter Lock)*** – такой специальный механизм, своеобразная блокировка, позволяющая только одному потоку управлять интерпретатором Python. Это означает, что в любой момент времени будет выполняться только один конкретный поток.

В Python существует такая библиотека, как **asyncio**, которая позволяет писать **асинхронный код** уже внутри **конкретного потока**. В чём смысл асинхронного кода: например, нам требуется отправить несколько сотен http-запросов и обработать их ответы. При синхронном коде, каждый запрос будет отправляться и обрабатываться последовательно – пока мы не дождёмся ответа от первого запроса (а это может быть очень долго), мы не отправим второй запрос и так далее, и время выполнения кода будет существенно увеличено.

При использовании асинхронного кода, запросы будут отправляться в любой «свободный» промежуток времени, т.е. пока мы отправили первый запрос, мы сможем отправить второй запрос, не дожидаясь ответа от первого, но позже уже сможем его обработать (как только придёт ответ)

В asyncio обычные функции заменяют «корутины» и объявляются как обычные функции, но перед объявлением ставится ключевое слово async.