

# Корутины

В терминологии библиотеки для работы с асинхронным кодом в Python [asyncio](#), «корутина» — это функция с ключевыми словами **async def** в определении, которую можно вызвать с ключевым словом **await**.

Простой вызов подобной функции без ключевого слова **await** вернет нам эту корутину, объект типа `coroutine`.

**Кооперативная многозадачность.** Один из видов многозадачности, в котором ОС не управляет переключением контекста выполнения. Вместо этого внутри одного процесса происходят периодические прерывания, или логическая передача управления другим подпрограммам.

**Корутина**, глобально — компонент компьютерной программы, применяемый для кооперативной многозадачности.

На первый взгляд, между тредом и корутиной совсем нет отличий. Обе сущности выполняются внутри одного процесса, и так же передается управление одной подпрограммы другой.

Но в случае с тредами ОС контролирует переключение подпрограмм, а в реализации корутин эти переключения контролирует сам Python.

Кооперативная многозадачность позволяет исполнять намного больше подпрограмм, чем треды. На число запущенных тредов в системе есть ограничение, и их инициализация требует больше ресурсов.

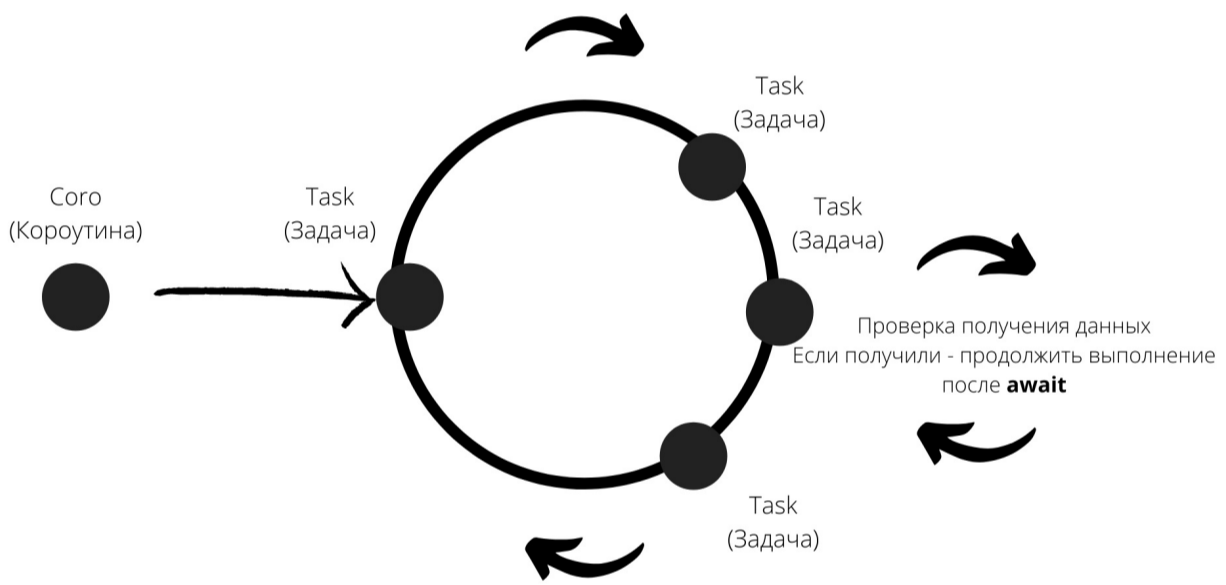
**Event loop.** Сущность, управляющая задачами в асинхронной программе. Следит за асинхронными задачами и принимает решение, какой задаче передать управление в определенный момент. Более подробно мы познакомимся с ней в следующей главе.

## Корутина отправляет асинхронный http-запрос:

```
async def main():
    async with aiohttp.ClientSession() as session:
        async with session.get('http://httpbin.org/get') as resp:
            print(resp.status)
            print(await resp.text())

loop = asyncio.get_event_loop()
loop.run_until_complete(main())
```

## Жизненный цикл корутины:



Сразу после вызова **await** перед корутиной создается задача и помещается в event loop. Она будет находиться в очереди до тех пор, пока не придет результат. При завершении IO-операции операционная система вызовет callback и поставит результат в **task**. Когда **event loop** дойдет до этой задачи, он поймет, что задача завершена, и передаст управление корутине. То есть программа продолжит свое выполнение с первоначального места вызова **await**.