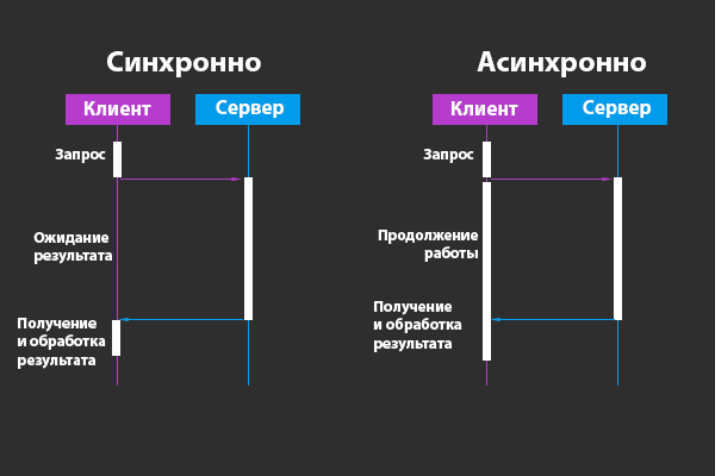
**Введение в асинхронное программирование**

**Асинхронное программирование**– это способ организации \*[кооперативной многозадачности](https://stepik.org/lesson/933699/step/1), который позволяет увеличить производительность и эффективность использования ресурсов. Это значит, что программа, написанная с использованием асинхронного программирования, может конкурентно выполнять несколько задач в рамках одного потока, в то время как в синхронном коде задачи выполняются последовательно одна за другой. Данный подход исключительно полезен в ситуациях, когда программа должна обрабатывать множество операций ввода/вывода, которые обычно занимают много времени.

**Синхронный запрос** ожидает получения ответа перед тем, как перейти к следующей задаче, то есть все запросы выполняются последовательно один за другим. Это может привести к "времени простоя", когда программа не делает ничего, ожидая ответа на запрос. Это может быть неэффективно, особенно если такое ожидание занимает много времени.

**Асинхронные запросы** позволяют программе продолжить выполнение других задач, пока текущая ожидает ответа на запрос. Это достигается за счет "переключения контекста". Когда программа выполняет асинхронный запрос, она может "отложить" эту задачу и перейти к следующей. Когда ответ на запрос получен, программа возвращается к этой задаче и продолжает ее выполнение.



Асинхронность может показаться сложной вначале, но с пониманием основных концепций и практикой она становится все проще. Важно понимать, что асинхронный код не может быть использован везде, и некоторые задачи лучше выполнять синхронно. Но в целом асинхронность может улучшить производительность и эффективность приложений, особенно в случаях, когда приложение должно обрабатывать много запросов.

Одним из ключевых понятий в асинхронном программировании является событийная модель. В событийной модели программа ждет таких событий, как входящие данные или завершение задачи, и выполняет необходимые действия, когда эти события происходят. Это означает, что ваше приложение постоянно слушает события и выполняет определенные действия в ответ на них. Это позволяет вашему приложению быть готовым к обработке новых событий в любое время, без необходимости ожидания завершения других задач.

### **Библиотека asyncio:**

В Python для асинхронного программирования есть встроенная библиотека **asyncio**. Для работы с данной библиотекой ее нужно просто импортировать.

import asyncio

В данном курсе будет разбираться библиотека asyncio, которая является основой для асинхронного программирования на Python, а также несколько других библиотек для решения специфических асинхронных задач.