Цикл событий – сердце asyncio. Именно цикл событий управляет тасками и их запуском

Корутина (сопрограмма) – объект, который возвращает асинхронная функция (async def). По сути, эта приписка к функции означает, что функцию можно запускать и останавливать в ивент лупе.

Таска – в таску оборачивается корутина. Именно таска отправляется в евент луп, просто корутину запихать не получится. Основные методы тасок – done(), который возвращает тру или фолз и results(), который вернёт возвращаемое значение функции-сопрограммы.

Фьючерс - объект ждущий результатов выполнения задачи. Объекты Future используются для соединения низкоуровневого кода, основанного на обратном вызове, с высокоуровневым кодом, основанном на синтаксисе async/await. По сути низкоуровневая штука. При разработке она не понадобится

Еще есть семафоры, очереди, ивенты, локи, и тд.

Генераторы и генераторные выражение

Генераторы генерируют возвращаемые значения, не имея всех значений. Они вычисляют нужные данные. Однако, нельзя вернуться обратно или пройтись по последовательности еще раз. Нужно будет создать объект-генератор еще раз.

Генераторные выражения возвращают также объект-генератор. Записываются в круглых скобках (списковые выражение – в квадратных).

Внутри генераторов используется ключевое слово yield, которые возвращает значение, но не заканчивает выполнение функции, а лишь приостанавливает.

**def** generator():  
   **yield** 'hello'  
   yield 'world'  
iterator = generator()

yield возвращает объект-генератор. Продолжится выполнение функции лишь после вызова next(iterator). Если дальше по коду в функции есть еще один yield, то снова вернётся объект-генератор. Если нет - вызовется исключение StopIteration.

Yield from