???? future ? awaitable

Практически редко применяются, но нужны для понимания

Будущие объекты можно использовать в выражениях await. Это означает «я посплю, пока в будущем объекте не будет установлено значение, с которым я могу работать, а когда оно появится, разбуди меня и дай возможность его обработать».

```
from asyncio import Future
my_future = Future()
print(f'my_future готов? {my_future.done()}')
my_future.set_result(42)
print(f'my_future готов? {my_future.done()}')
print(f'Какой результат хранится в my_future? {my_future.result()}')
```

Вариант для использования не до конца определенной переменной:

```
from asyncio import Future
import asyncio
def make_request() -> Future:
    future = Future()
    asyncio.create_task(set_future_value(future))
    return future
async def set_future_value(future) -> None:
    await asyncio.sleep(1)
    future.set_result(42)
async def main():
    future = make_request()
    print(f'Будущий объект готов? {future.done()}')
    value = await future
    print(f'Будущий объект готов? {future.done()}')
    print(value)
asyncio.run(main())
```

Любой объект, который реализует метод __await__, можно использовать в выражении await. Это объекты, допускающие ожидание (Awaitaible).

Revision #1
Created 5 May 2025 07:54:39 by Admin
Updated 24 June 2025 14:28:25 by Admin