



Асинхронная функция может быть приостановлена и возобновлена в процессе выполнения, позволяя тем временем выполнять другие задачи.

```
import time

def fun1(x):
    print(x**2)
    time.sleep(2)
    print('fun1 завершена')

def fun2(x):
    print(x**3)
    time.sleep(2)
    print('fun2 завершена')

def main():
    fun1(4)
    fun2(4)

print(time.strftime('%X'))
main()
print(time.strftime('%X'))
```

# Вывод

```
16:17:35
16
fun1 завершена
2.0
fun2 завершена
16:17:41
```

# Воспользуемся asyncio

```
import asyncio
import time
async def fun1(x):
    print(x**2)
    await asyncio.sleep(2)
    print('fun1 завершена')
async def fun2(x):
    print(x**3)
    await asyncio.sleep(2)
    print('fun2 завершена')

async def main():
    task1 = asyncio.create_task(fun1(4))
    task2 = asyncio.create_task(fun2(4))

    await task1
    await task2

print(time.strftime('%X'))
asyncio.run(main())
print(time.strftime('%X'))
```

16:02:49

16

2.0

fun1 завершена

fun2 завершена

16:02:52