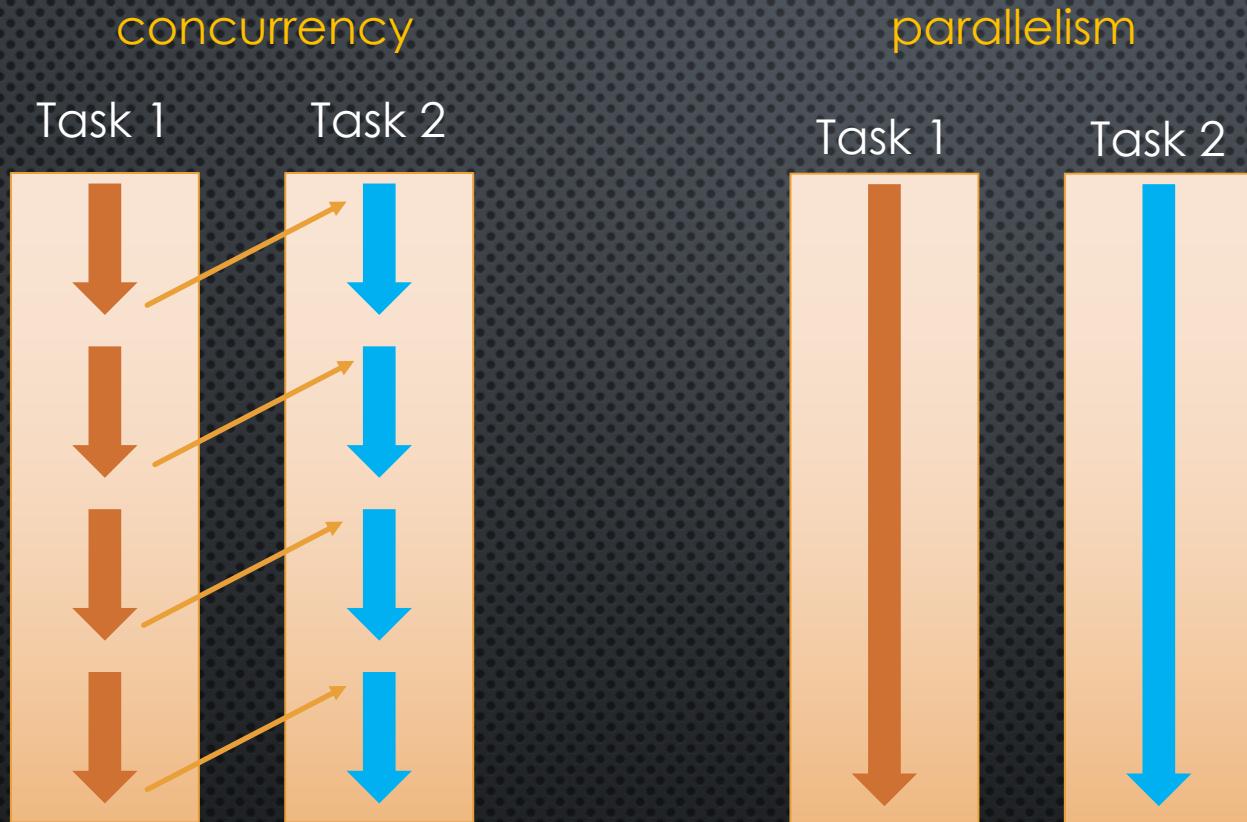
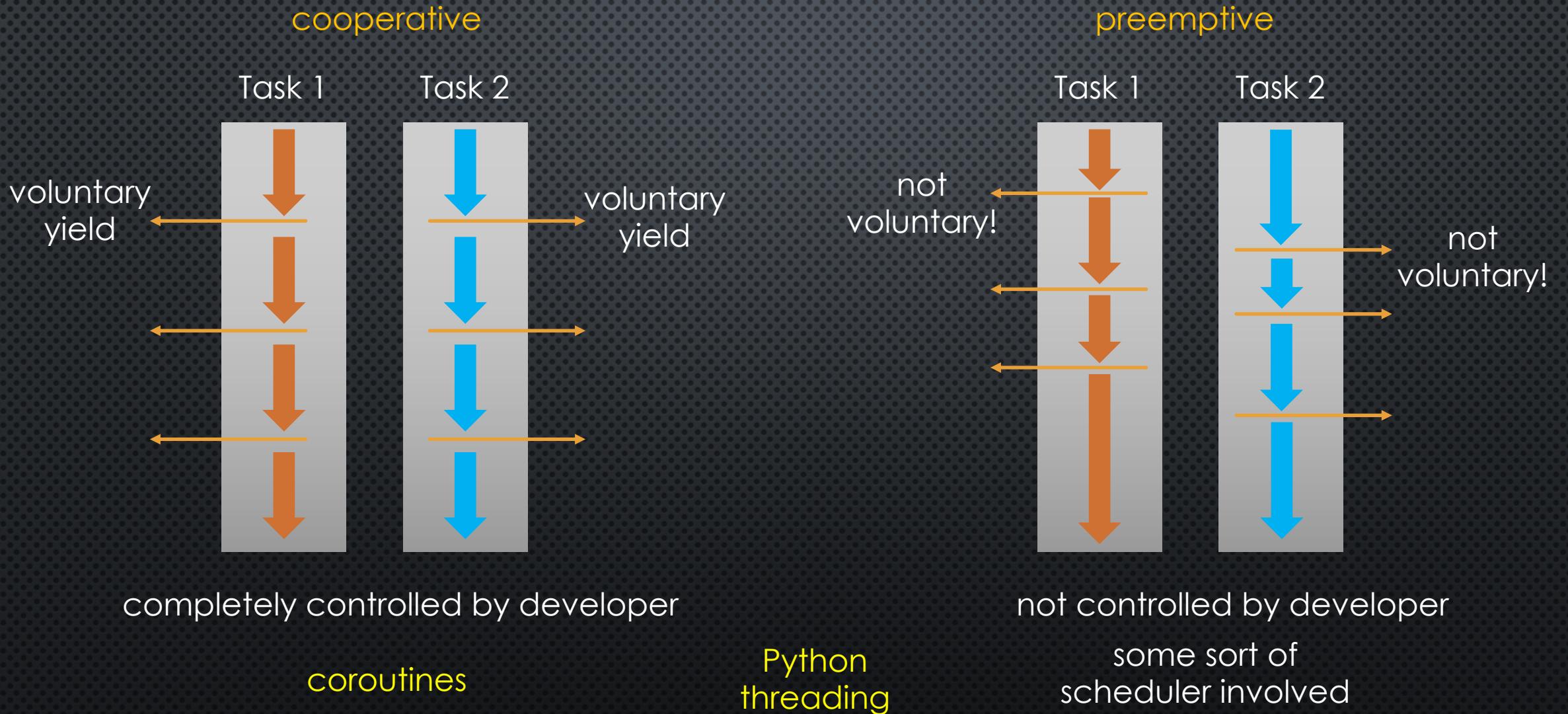


GENERATORS AS COROUTINES

Concurrency vs Parallelism



Cooperative vs Preemptive Multitasking



Coroutines

Cooperative multitasking

Concurrent, not parallel

→ Python programs execute on a "single thread"

Global Interpreter Lock → GIL

Two ways to create coroutines in Python

→ generators

→ uses extended form of `yield`

→ recent addition: `asyncio`

→ native coroutines

→ uses `async / await`

This section is **not** about

asyncio

native coroutines

threading

multiprocessing

→ parallelism

This section is **about**

learning the basics of generator-based coroutines

some practical applications of these coroutines