Concurrency Models in Python — Overview

Why multiple concurrency models exist

Programs are slow primarily because they wait. Python supports multiple concurrency models to overlap waiting or to execute work in parallel, depending on the nature of the workload.

Target Usage

```
# I/O-bound → threads or async
# CPU-bound → processes
```

Coding Problem

Choose an appropriate concurrency model for a given workload and understand the trade-offs of each option.

Concurrency Models

Threads overlap waiting but do not provide true CPU parallelism. Async uses a single-threaded event loop to scale I/O efficiently. Processes provide true parallelism at the cost of higher overhead.

Observed Effect

Choosing the wrong model leads to wasted resources and worse performance. Correct model selection improves throughput and predictability.

Key Insight

Concurrency is about making progress while waiting; parallelism is about doing work simultaneously. Python supports both through different mechanisms.