Spark Execution Timeline

0. Driver(lazy)

- `parallelize` → RDD + lineage
- `mapPartitions` → adds transform
- `createDataFrame` → logical plan only

! Spark may **peak at a row** for schema inference

1. Action triggered (`show`, `count`, `write`)

- Scheduler executes lineage
- 2 partitions → 2 tasks → 1 stage

2. Executor startup

For each partition:

- JVM executor launches (or reuses) Python worker
  - Worker is reused when it finished current partition, and receives the next partition from the scheduler
- Partition elements streamed into Python iterator
- `f(records_iter)` called **once**

3. Generator runs

- Each `yield`:
  - Emits one row
  - Immediately serialized back to JVM
- Generator continues until exhausted

4. JVM conversion

- Python `Row` / tuple → pickled
- JVM converts to `InternalRow` / `UnsafeRow`

5. DataFrame materialization

- RDD of rows backs the DataFrame
- Stage completes when all partitions finish

6. Action completes

- `show()` → collect limited rows
- `count()` → aggregate counts
- `write()` → write partition outputs