Objective:

Explain what constitutes “good code” from a Senior Data Engineer perspective

and identify authoritative external sources that validate these principles.

Scope:

- Focus on production-grade Python data engineering
- Assume ELT pipelines, layered architectures, and long-term maintenance
- Target audience: senior engineers, recruiters, architects

Deliverables:

Part 1 – Definition of Good Code

Explain what makes code “good” in professional data engineering, structured by:

- Correctness & determinism
- Readability & cognitive load
- Architectural separation of concerns
- Robustness & failure handling
- Observability & reproducibility
- Testability & maintainability
- Performance (contextualized, not premature)
- Security & governance
- Documentation & decision traceability

Part 2 – Validation Sources

List authoritative sources that validate the above principles.

For each source include:

- Name
- Organization or origin
- What aspect of good code it validates

- Why it is credible in professional contexts

Constraints:

- Use engineering language
- Avoid beginner-level advice
- No emojis, no filler text
- Do not invent sources