Prompt Engineering

Youssef Guba

What is the Prompt Engineering?

Why?

- **Cost control** – A well-engineered prompt can cut tokens 30-60 %, lowering inference bills.
- **Personalisation** – Dynamic templating (e.g., user locale, reading level) at render-time.
- **Safety & compliance** – Guardrail prompts help enforce style guides, content policies, PII masking.

Ways of Prompt Engineering

- Instructional (Zero-shot)
- Few-shot / Exemplar-based
- Constrained / Output-structured
- Open-ended / Creative
- Role/Persona
- Chain-of-Thought (CoT)
- Self-Reflection / Self-Critique
- Tell him Create the Prompt

Best Practices

- **Clarity over cleverness** – Short, declarative sentences outperform flowery prose.
- **Give context early** – Domain, target audience, constraints.
- **Strong delimiters** – Triple-backticks,
- **Specify output format unambiguously** – Particularly for programmatic consumption.
- **Iterative decomposition** – Break a complex job into smaller chained prompts for higher reliability.

Advanced Techniques

- **Multi-agent Systems**: Specialized personas negotiating; eg. Planner-Coder-Tester loop.
- **Program-of-Thought**: Model writes code, your sandbox executes, model incorporates results.
- **Differential Prompting**: A/B prompts against each model version;

Prompt Chaining

Benefits

- **Decomposition** reduces per-step cognitive load on the model.
- **Observability**: you can log/trace each node.

Cost math: 3 small prompts with focused context often < 1 monolithic monster prompt.

General Tools

- ChatGPT
- DeepSeek
- Gemini
- Grok
- Manus
- Claude

Specific Tools

- Kling
- Google Veo
- Runway
- Lovable
- bolt
- Cursor
- Github Copilot
- Stitch

Don’t forget to use your mind!