

Simple Engineering

Prompt Engineering Cheat Sheet

Four spices for prompts that work INTENT → CONTEXT → OUTCOME → REFERENCES

THE PROMPT SPICES



M365 Copilot user Outlook · Word · Teams · Excel



Developer · GitHub Copilot in the IDE

1 INTENT

What do you want?
A verb + the exact deliverable.

- ✗ *Help me with these sales numbers.*
- ✓ Make a 5-row table of Q2 sales by region.

- ✗ *fix this*
- ✓ Refactor parseConfig() to return Result<Config, Error>.

2 CONTEXT

What must it know?
Who, audience, situation, why.

- ✗ *Write a project update.*
- ✓ Update for my director, non-technical.
We are 2 weeks behind.

- ✗ *Write a caching function.*
- ✓ Add an LRU cache (max 500).
Node 20, TypeScript strict.

3 OUTCOME

What is “done”?
Format, length, tone, limits.

- ✗ *Summarise this document.*
- ✓ 5 bullets, under 150 words, professional tone.

- ✗ *Add some tests.*
- ✓ Edge tests for empty, null & 429-retry.
Reuse test-utils.

4 REFERENCES

What to rely on?
The files/links, and ask it to cite.

- ✗ *What’s our travel policy?*
- ✓ From the attached HR policy only.
State the limit and cite it.

- ✗ *Add a rate limiter.*
- ✓ Follow utils/rateLimiter.ts + CONTRIBUTING.md.

ALSO TRY these spices

assign a role · give one example · break into numbered steps · say what to avoid · ask it to clarify first · iterate, do not restart.

VALIDATE

verify facts & figures · keep sensitive data appropriate · you own the output.

GCES = Good Copilots Everywhere Succeed