

Claude Code Power Prompts

Free sample - 5 prompts

Last updated: 2026-05-19

This is the free 5-prompt sample of Claude Code Power Prompts. The paid pack has 30 prompts across six categories: code review, debugging, testing, architecture, documentation, and devops. Every prompt is plain text - paste into Claude Code, claude.ai, or any Claude interface.

What's inside the paid pack

- 30 copy-paste prompts in an 8-page PDF.
- TIP box for every prompt - when to use, when not to use.
- Designed to paste into CLAUDE.md - Claude Code auto-loads them every session.
- Instant Stripe-verified download. 7-day no-questions refund - reply to receipt.

Three checkout options

- GBP 3 - launch (first 10 buyers) - same PDF
- GBP 5 - standard one-time price
- GBP 1+ - pay what you can (from one pound)

Try the full pack

Buy: <https://claude-code-examples.vercel.app/buy/> | See all 30:
<https://claude-code-examples.vercel.app/all-30-prompts/>

*Built in public. Licensed CC BY 4.0 - attribute claude-code-examples.vercel.app.
Hosted on Vercel. Refund within 7 days - reply to your Stripe receipt.*

The 5 free prompts

Updated 2026-05-19 - copy-paste into Claude Code, claude.ai, or the Anthropic API.

01. Security Audit

Review this file for security vulnerabilities, focusing on OWASP Top 10. List each issue with severity (critical/high/medium/low), the exact line number, why it's dangerous, and a concrete fix.

TIP: Works best on auth, payment, and data-handling code. Paste the file content immediately after the prompt.

02. Stack Trace Autopsy

Given this stack trace, trace the execution path that produced it. For each frame: which file/function, what state would cause it to throw, and the 3 most likely root causes ranked by probability. End with one diagnostic command I can run to confirm the top cause.

TIP: Use when you have a stack trace but no working theory yet.

03. Coverage Risk Ranker

Look at this code and our current test coverage. Identify the 20% of code most likely to cause production bugs (high complexity, weak tests, recent changes, business-critical paths). For each, give me one specific test I could add right now that would meaningfully reduce risk.

TIP: Best after a big merge. Surfaces gaps your coverage % hides.

04. Feature Addition - 3 Approaches

I want to add [FEATURE] to this codebase. Propose three implementation approaches. For each: a one-paragraph description, the files that change, complexity 1-5, risk 1-5, and one downside the others don't have. Force at least one non-obvious option - not just incremental variations.

TIP: The forced 'one non-obvious option' is the gold here. Use before any non-trivial feature.

05. Hook-First README

Rewrite this README's first 5 lines so a developer scanning it for 5 seconds either commits to using the tool or leaves. Then rewrite the 'getting started' section so a working example is visible within 30 seconds of arrival. Show me before/after.

TIP: Use on any project README older than 6 months.