

50 PROMPTS. 8 CATEGORIES. 5 WORKFLOWS.

The Claude Prompt Pack

High-Value Prompts to Get More Done With Claude

Copy-paste prompts that turn Claude into a debugging partner, a study coach, a code reviewer, a career strategist, and a thinking partner — built for people who want outcomes, not explanations of how prompting works.

haas.dev

Engineering mindset over tutorials.

Built for Claude specifically

Every prompt in this pack is written to work inside Claude — taking advantage of what it's actually good at: reasoning through a problem step by step, holding a lot of context (long documents, full codebases, messy notes), and following detailed multi-step instructions without losing the thread.

This is a toolkit, not a book. Skim the category pages, find the prompt that matches what you're stuck on right now, copy it, fill in the brackets, and go. Come back next time you hit a different kind of problem.

- Replace every [PLACEHOLDER] before sending — an unfilled bracket is the #1 reason a good prompt gives a generic answer.
- Paste real material (your actual code, notes, resume, document) instead of describing it. Claude works from source material far better than from a summary of it.
- These prompts are starting points, not one-shot magic. Follow up inside the same conversation — “go deeper on point 2” — instead of restarting.

WHAT'S INSIDE

- 1. Learning & Studying — 6 prompts
- 2. Coding & Development — 8 prompts
- 3. Career & Jobs — 6 prompts
- 4. Research — 6 prompts
- 5. Productivity — 6 prompts
- 6. Writing & Content — 6 prompts
- 7. Business & Entrepreneurship — 6 prompts
- 8. AI Workflows — 6 prompts
- Bonus — 5 combined workflows

CATEGORY 1

Learning & Studying

For turning confusion into understanding — and understanding into something that sticks.

PROMPT #1

The Concept Diagnostician

What it does: Pinpoints exactly what you're misunderstanding about a concept, explains it with a simple mental model, then tests you without giving away the answers.

Best for: *Anyone stuck on a specific topic and not sure exactly what part they don't understand*

COPY & PASTE INTO CLAUDE

```
Act as my learning coach for [SUBJECT/TOPIC].

I'm struggling with: [SPECIFIC CONCEPT]
My current understanding (in my own words): [WHAT YOU THINK YOU KNOW]
My experience level: [EXPERIENCE LEVEL]

Do this in order:
1. Ask me 1-2 diagnostic questions to find out exactly where my understanding breaks down – don't explain anything yet.
2. Based on my answers, explain the concept using one simple, concrete mental model or analogy.
3. Walk me through one example, then a slightly harder example.
4. Test me with 3 questions of increasing difficulty. Do not reveal the answers – wait for my response to each one before giving feedback and moving to the next.
```

CUSTOMIZE THESE

Set the subject, the specific concept (not the whole topic), and your honest current understanding — vague input here produces a generic explanation instead of a diagnosis.

WHAT CLAUDE SHOULD GIVE YOU

1-2 diagnostic questions, a plain explanation once it knows your gap, worked examples, and a 3-question test that only reveals answers after you respond.

PRO TIP

Answer the diagnostic questions honestly, even “I don't know” — that's exactly the information Claude needs to target the explanation.

PROMPT #2

The Active Recall Quiz Builder

What it does: Converts your raw notes into a self-testing quiz sequence instead of a summary you'll passively re-read.

Best for: *Students turning class or reading notes into study material*

COPY & PASTE INTO CLAUDE

```
Act as a study coach who builds active-recall quizzes from notes.

Here are my notes on [TOPIC]:

[PASTE NOTES]

Build a quiz from this material:
1. 8-10 questions ordered from foundational to harder, covering only what's actually in my notes.
2. Mix formats: a few short-answer, a few "explain this in your own words," and 1-2 applied/scenario questions.
3. Ask me one question at a time. Wait for my answer before giving feedback and the next question.
4. After each answer, tell me if I'm right, what I missed if I'm wrong, and correct any misconception — don't just say "correct/incorrect."
5. At the end, tell me which 2-3 topics from my notes I'm weakest on so I know what to review again.
```

CUSTOMIZE THESE

Paste your actual notes, not a topic name — the quiz should test what you were taught, not a generic version of the subject.

WHAT CLAUDE SHOULD GIVE YOU

A one-at-a-time quiz session with real feedback per answer, and a final summary of your weakest 2-3 areas to revisit.

PRO TIP

Run this again a few days later on the same notes without looking at your old answers — the gap between the two attempts shows you what actually stuck.

PROMPT #3

The Study Plan Failure Diagnostician

What it does: Analyzes a study plan that isn't working and identifies the real reason — not just “try harder.”

Best for: *Anyone whose study plan isn't working and doesn't know why*

COPY & PASTE INTO CLAUDE

```
Act as a learning strategist who diagnoses why study plans fail.

My goal: [GOAL, e.g. pass a certification / learn X by a date]
My plan: [DESCRIBE YOUR CURRENT PLAN/SCHEDULE]
What's actually happening: [WHAT'S GOING WRONG – e.g. skipping sessions, forgetting material,
running out of time]

Diagnose this honestly:
1. Identify the most likely root cause – distinguish between a motivation problem, a design
problem (plan is unrealistic or poorly sequenced), and a method problem (passive studying instead
of active recall).
2. Point out anything in the plan that was doomed from the start, and why.
3. Rebuild the plan around the real constraint, not the ideal version of my week.
4. Give me one small change to test this week before overhauling everything.

Be direct. Don't just tell me to "stay consistent" – explain specifically what's misaligned.
```

CUSTOMIZE THESE

Describe what's actually happening, not what should be happening — the honest gap between the plan and reality is what gets diagnosed.

WHAT CLAUDE SHOULD GIVE YOU

A specific root-cause diagnosis (motivation vs. design vs. method), a rebuilt plan around your real constraints, and one small testable change for this week.

PRO TIP

Report back after a week with what actually happened — treat this as an ongoing diagnostic loop, not a one-time fix.

PROMPT #4

The Exam Blueprint Builder

What it does: Turns a syllabus or exam scope into a realistic, prioritized prep plan instead of “study everything equally.”

Best for: *Students preparing for a specific exam or certification with limited time*

COPY & PASTE INTO CLAUDE

```
Act as an exam prep strategist.

Exam/certification: [EXAM NAME]
Scope/syllabus: [PASTE SYLLABUS, TOPIC LIST, OR EXAM OUTLINE]
Time until exam: [TIMEFRAME]
Hours available per week: [HOURS]
My current strength/weakness by topic (if known): [NOTES, OR "UNKNOWN"]

Build a prep blueprint:
1. Rank the topics by likely weight on the exam and by how foundational they are to other topics.
2. Flag which topics are highest-leverage to master first given my timeframe.
3. Build a week-by-week schedule that fits my available hours, including built-in review sessions (not just first-pass learning).
4. Recommend a practice-testing cadence – when I should start doing full practice exams, not just topic review.
5. Tell me honestly if my timeframe is unrealistic for the scope, and what to cut if so.
```

CUSTOMIZE THESE

Paste the real syllabus or topic list if you have one — a vague exam name alone forces Claude to guess at scope.

WHAT CLAUDE SHOULD GIVE YOU

A prioritized, week-by-week prep schedule fit to your real hours, a practice-exam cadence, and an honest readiness check on your timeframe.

PRO TIP

Use The Active Recall Quiz Builder (Prompt #2) each week on that week's topics to actually test what the blueprint says you should know by then.

PROMPT #5

The Notes-to-Learning-System Converter

What it does: Turns raw, messy notes into a structured learning system — organized by concept, with gaps flagged, not just cleaned up formatting.

Best for: *Anyone with messy, scattered notes who wants an actual study system, not just a cleaner document*

COPY & PASTE INTO CLAUDE

```
Act as a learning systems designer.
```

```
Here are my raw notes on [TOPIC]:
```

```
[PASTE MESSY NOTES]
```

```
Turn this into a learning system, not just a cleaner document:
```

1. Group the content into core concepts (not just the order I wrote it in).
2. For each concept, write a one-line plain-English summary I could use for quick review.
3. Identify any gaps – places where my notes reference something without explaining it, or where a concept seems incomplete.
4. Suggest 3-5 questions I should be able to answer if I actually understand this material, without answering them yourself.
5. Recommend an order to review these concepts in, based on which ones build on others.

CUSTOMIZE THESE

Paste your actual raw notes, typos and disorganization included — the value is in Claude finding structure you couldn't see, not in you pre-organizing them first.

WHAT CLAUDE SHOULD GIVE YOU

A concept-grouped structure with one-line summaries, explicitly flagged knowledge gaps, self-check questions, and a recommended review order.

PRO TIP

Once you have the structured version, feed each concept into The Concept Diagnostician (Prompt #1) for anything you can't confidently explain back.

PROMPT #6

The Socratic Tutor

What it does: Teaches a topic entirely through guided questions instead of direct explanations, forcing you to reason your way to the answer.

Best for: *Learners who want to build real understanding instead of collecting answers*

COPY & PASTE INTO CLAUDE

```
Act as a Socratic tutor teaching me [TOPIC].

My starting knowledge: [EXPERIENCE LEVEL]
What I want to understand by the end: [GOAL]

Rules for this session:
1. Do not explain concepts directly. Teach entirely through questions that guide me toward the answer myself.
2. Ask one question at a time and wait for my response before continuing.
3. If I answer incorrectly, don't correct me directly – ask a follow-up question that helps me see the flaw in my own reasoning.
4. If I'm completely stuck after two attempts on the same question, give me a small hint, not the answer, and let me try again.
5. Periodically summarize what I've established so far, in my own words, to confirm I'm actually building understanding and not just guessing your way to yes.

Start with your first question.
```

CUSTOMIZE THESE

Set a specific topic and be honest about your starting knowledge — this method is slower than a direct explanation, so it works best on concepts worth truly understanding, not quick facts.

WHAT CLAUDE SHOULD GIVE YOU

A pure question-driven dialogue that only gives hints (never direct answers) when you're stuck, with periodic checkpoints summarizing what you've built up yourself.

PRO TIP

Use this for concepts you'll need to reason about under pressure later (interviews, exams, debugging) — it builds the reasoning skill, not just the answer.

CATEGORY 2

Coding & Development

For understanding, improving, and building software — using Claude's ability to reason through code and hold a lot of context at once.

PROMPT #7

The Codebase Architecture Mapper

What it does: Gives Claude a codebase (or a large chunk of one) and has it map out the architecture, key components, and how data flows — the orientation a new team member usually gets from a teammate.

Best for: *Developers joining a new codebase, or revisiting an old one, who need the big picture fast*

COPY & PASTE INTO CLAUDE

```
Act as a senior engineer onboarding me onto an unfamiliar codebase.

Here is the code / file structure I'm working with:

[PASTE CODE, FILE TREE, OR MULTIPLE FILES]

What this project is (if known): [BRIEF CONTEXT]

Map this out for me:
1. High-level architecture – what are the major components/modules and what does each one own?
2. How data/requests flow through the system, start to finish, for a typical operation.
3. The 3-5 files or modules most central to understanding this codebase – where I should focus first.
4. Any patterns, conventions, or architectural decisions worth knowing before I start editing (naming conventions, state management approach, etc.).
5. Anything that looks unusual, fragile, or worth extra caution before touching it.

Present this as an onboarding brief, not a line-by-line code walkthrough.
```

CUSTOMIZE THESE

Paste as much real code/structure as you reasonably can — this prompt is built for Claude's ability to hold large amounts of context, so more real material produces a sharper map.

WHAT CLAUDE SHOULD GIVE YOU

An onboarding-style brief: architecture overview, a data-flow walkthrough, the highest-priority files to understand first, conventions to follow, and fragile areas to be careful with.

PRO TIP

Follow up with “now trace what happens end-to-end when [SPECIFIC ACTION] occurs” to turn the static map into a concrete walkthrough of the exact flow you need to work on.

PROMPT #8

The Senior Engineer Code Reviewer

What it does: Reviews your code the way an experienced senior engineer would — correctness, readability, structure, performance, and security, with reasoning for every flag, not just a style pass.

Best for: *Developers who want a thorough review before opening a pull request*

COPY & PASTE INTO CLAUDE

```
Act as a senior [TECH STACK] engineer doing a code review before this goes to production.
```

```
Code:  
[PASTE CODE]
```

```
What this code is supposed to do: [GOAL]  
Constraints (style guide, must not break X, performance requirements): [CONSTRAINTS]
```

```
Review in this order, and only flag things that actually matter:  
1. Correctness – logic errors, missed edge cases, incorrect assumptions.  
2. Readability – anything a teammate would need to re-read twice to understand.  
3. Structure – naming, function boundaries, and maintainability issues.  
4. Performance – only flag if it matters at realistic scale for this code.  
5. Security – input validation, injection risk, exposed secrets.
```

```
For each issue: the exact line/section, why it matters, and the corrected code.  
Close with a clear verdict: ready to merge, or what must change first.
```

CUSTOMIZE THESE

Paste the real code and be specific about constraints — a style guide or performance requirement changes what counts as an issue worth flagging.

WHAT CLAUDE SHOULD GIVE YOU

A categorized list of issues with reasoning and corrected code for each, plus a clear merge-ready verdict at the end.

PRO TIP

Ask Claude to only fix specific flagged issues and return the full updated file, instead of pasting the whole file back into a new request.

PROMPT #9

The Unfamiliar Code Explainer

What it does: Explains what a piece of unfamiliar code actually does, section by section, in plain language — built for code you need to work with fast, not admire.

Best for: *Developers who need to quickly understand code someone else wrote*

COPY & PASTE INTO CLAUDE

```
Act as a senior engineer explaining unfamiliar code to a teammate.

Code:
[PASTE CODE]

My experience level: [EXPERIENCE LEVEL]
Why I need to understand this: [GOAL, e.g. fixing a bug in it / extending it / reviewing it]

Explain it in this order:
1. One-paragraph summary of what this code does overall.
2. Section-by-section breakdown of the logic, in plain language – skip parts that are self-explanatory.
3. Anything non-obvious: side effects, assumptions the code is making, or dependencies on external state.
4. Given why I need to understand this, tell me specifically which part I should focus on and why.
```

CUSTOMIZE THESE

Paste the actual code and say specifically why you need to understand it — that context changes which parts of the explanation actually matter to you.

WHAT CLAUDE SHOULD GIVE YOU

A quick overall summary, a plain-language section breakdown skipping the obvious parts, flagged non-obvious behavior, and a pointer to exactly what matters for your stated goal.

PRO TIP

If the file is large, paste it in logical chunks (one function/class at a time) and ask Claude to hold the running context across your messages.

PROMPT #10

The Error Root-Cause Diagnostician

What it does: Diagnoses an error by reasoning through likely causes systematically, instead of guessing at a fix on the first read.

Best for: *Developers debugging an error they don't immediately understand*

COPY & PASTE INTO CLAUDE

```
Act as a debugging partner who reasons through problems systematically rather than guessing.

Error:
[PASTE FULL ERROR/STACK TRACE]

Context:
- Stack: [TECH STACK]
- What I was doing when it happened: [GOAL/ACTION]
- Relevant code: [PASTE CODE]
- What I've already tried: [ATTEMPTS, OR "NOTHING YET"]

Work through this systematically:
1. List the 2-3 most likely root causes, ranked by probability given the context.
2. For the top cause, explain the reasoning — what in the error/code points to it.
3. Give the specific fix.
4. Tell me how to confirm the real cause if the top guess turns out to be wrong.
5. Suggest one check I could add so this fails loudly and clearly next time instead of confusingly.
```

CUSTOMIZE THESE

Include what you've already tried — it prevents Claude from re-suggesting a fix you've ruled out and speeds up getting to the real cause.

WHAT CLAUDE SHOULD GIVE YOU

A ranked list of likely causes with reasoning (not just one guess), a specific fix, a way to confirm you're right, and a suggestion to make the failure clearer next time.

PRO TIP

If the first suggested cause doesn't hold up, tell Claude exactly what you observed when you tested it — that new data point usually narrows it down fast.

PROMPT #11

The Refactor Planner

What it does: Plans a refactor of existing code — what to change and in what order — without altering behavior, prioritized by risk and impact.

Best for: *Developers who know code needs cleaning up but aren't sure where to start without breaking things*

COPY & PASTE INTO CLAUDE

```
Act as a senior engineer planning a refactor.

Code:
[PASTE CODE]

Why it needs refactoring: [REASON, e.g. hard to extend / duplicated logic / hard to test]
Constraint: behavior must not change.

Plan the refactor:
1. Identify the specific structural problems (not style nitpicks) – duplication, tight coupling, unclear responsibility boundaries, etc.
2. Propose a refactor plan broken into small, safe steps, ordered so each step is independently testable.
3. For the highest-priority step, show the actual refactored code.
4. Flag anything in the current code that makes this refactor riskier than it looks (hidden dependencies, implicit behavior).
5. Suggest what to test after each step to confirm behavior didn't change.
```

CUSTOMIZE THESE

State the real reason it needs refactoring — “hard to test” and “hard to extend” lead to different priority orderings.

WHAT CLAUDE SHOULD GIVE YOU

A structural problem list, a step-by-step refactor plan ordered by safety, actual rewritten code for the first step, and a testing checklist per step.

PRO TIP

Do one step at a time in the same conversation and confirm your tests pass before asking for the next — refactoring in one giant leap is where bugs hide.

PROMPT #12

The Test Suite Generator

What it does: Generates tests for existing code that actually target realistic edge cases and failure modes, not just the happy path.

Best for: *Developers who need meaningful test coverage for existing code, not just line-count padding*

COPY & PASTE INTO CLAUDE

```
Act as a senior engineer writing tests for this code.

Code:
[PASTE CODE]

Testing framework: [FRAMEWORK, e.g. Jest, pytest]
What matters most for this code: [CONSTRAINT, e.g. correctness under bad input / performance / concurrency]

Generate tests that:
1. Cover the happy path first, briefly.
2. Focus most of the effort on realistic edge cases: empty/null input, boundary values, unexpected types, and failure conditions specific to what this code does.
3. Include at least one test that would have caught a plausible real bug in code like this.
4. Are named descriptively enough that a failing test tells you what broke without opening the file.

After the tests, list any edge cases you think are worth testing but couldn't write with the information given, and what you'd need to know to write them.
```

CUSTOMIZE THESE

State what matters most for this specific code — it changes whether the generated tests emphasize input validation, timing, concurrency, or something else.

WHAT CLAUDE SHOULD GIVE YOU

A test file weighted toward meaningful edge cases over happy-path padding, descriptive test names, and a list of untested-but-worth-testing cases with what's missing to write them.

PRO TIP

Ask Claude to also write one test that currently fails against a bug you suspect exists — it's a fast way to confirm a hunch before you go hunting through the code.

PROMPT #13

The New Technology Ramp-Up Guide

What it does: Builds a fast, targeted ramp-up plan for a new technology based on what you already know, instead of a generic “beginner tutorial.”

Best for: *Developers who need to learn a new framework or tool quickly, using what they already know*

COPY & PASTE INTO CLAUDE

```
Act as a senior engineer who already knows [NEW TECHNOLOGY] and is onboarding me onto it.

What I already know: [EXISTING SKILLS/STACK]
Why I'm learning this: [GOAL, e.g. new job requirement / side project]
Timeframe: [TIMEFRAME]

Build my ramp-up plan:
1. Map [NEW TECHNOLOGY] concepts to equivalent concepts in what I already know, wherever a real equivalent exists – skip this where there isn't one.
2. Identify the 20% of this technology that will cover 80% of what I'll actually use day to day.
3. Flag the parts that work differently from what I'm used to and are common sources of mistakes for people with my background specifically.
4. Suggest one small real project (not a toy tutorial) that would force me to use the core concepts.
5. List 2-3 things I should deliberately NOT worry about learning yet.
```

CUSTOMIZE THESE

Be specific about what you already know — the analogies and the “what's different” warnings are the most valuable part, and both depend on your real background.

WHAT CLAUDE SHOULD GIVE YOU

A background-relative crash course: concept mappings to what you know, the highest-leverage 20%, background-specific pitfalls, one real project idea, and what to skip for now.

PRO TIP

Once you pick the project, come back and use The Codebase Architecture Mapper (Prompt #7) on your own code as you build it, to sanity-check your structure early.

PROMPT #14

The Documentation Untangler

What it does: Turns confusing official documentation into a working example tailored to what you're actually trying to do.

Best for: *Developers stuck on confusing or sparse official documentation*

COPY & PASTE INTO CLAUDE

```
Act as a senior engineer who's used [LIBRARY/TOOL/API] extensively.

Here's the documentation I'm stuck on:

[PASTE DOCS SECTION OR LINK CONTENT]

What I'm actually trying to accomplish: [GOAL]
My current attempt (if any): [PASTE CODE]

Help me get unstuck:
1. Explain in plain language what this part of the documentation is actually saying, including anything it's assuming I already know.
2. Give me a minimal working example aimed specifically at my stated goal, not the generic example from the docs.
3. If my current attempt is close, tell me exactly what's wrong with it.
4. Flag any common gotchas with this specific library/feature that the docs don't make obvious.
```

CUSTOMIZE THESE

Paste the actual documentation text, not just a link — and be specific about your goal so the example is aimed at your use case instead of the generic docs example.

WHAT CLAUDE SHOULD GIVE YOU

A plain-language translation of the docs, a working example targeted at your actual goal, a specific fix if you're already close, and common gotchas the docs left out.

PRO TIP

If the docs example uses different terminology than you expect, explicitly ask Claude to map the docs' terms to the concepts you already know from other tools.

CATEGORY 3

Career & Jobs

For getting past the resume screen, closing skill gaps, and walking into interviews prepared.

PROMPT #15

The Resume Rejection Diagnostician

What it does: Analyzes your resume the way an overworked recruiter skimming for 7 seconds would, and identifies the specific reasons it's getting passed over.

Best for: *Job seekers getting rejected or ghosted without knowing why*

COPY & PASTE INTO CLAUDE

```
Act as a recruiter who screens resumes for [TARGET ROLE] roles and has to make a keep/reject decision in under 10 seconds.

Resume:
[PASTE RESUME TEXT]

Target role: [TARGET ROLE]

Give me an honest screening reaction:
1. What's the very first impression in the first 3 lines – does it immediately signal I'm a fit for [TARGET ROLE], or does a recruiter have to work to figure that out?
2. What would make a recruiter skim past this resume without reading further?
3. Identify anything vague, generic, or duty-based (rather than achievement-based) that weakens it.
4. Point out any obvious gaps or red flags a recruiter would silently note.
5. Give me the top 3 changes that would most improve my odds, ranked by impact.

Be honest, not encouraging – I'd rather know now than keep getting rejected silently.
```

CUSTOMIZE THESE

Paste the actual resume text and a specific target role — “any job” produces vague feedback because fit depends on the role.

WHAT CLAUDE SHOULD GIVE YOU

A blunt first-impression reaction, what causes a skim-past, flagged vague/duty-based lines, noted red flags, and a ranked top-3 fix list.

PRO TIP

Run this against 2-3 different target roles if you're applying broadly — the same resume often has different weak points depending on the role.

PROMPT #16

The Resume-to-Job-Description Matcher

What it does: Compares your resume directly against a specific job description and shows exactly where you match, where you're weak, and what language to add.

Best for: *Job seekers applying to a specific role who want to know their real odds and gaps*

COPY & PASTE INTO CLAUDE

```
Act as a hiring manager comparing a resume against a specific job description.

Job description:
[PASTE JOB DESCRIPTION]

My resume:
[PASTE RESUME]

Compare them:
1. List the job description's core requirements, split into must-have and nice-to-have.
2. For each must-have, tell me whether my resume clearly demonstrates it, weakly implies it, or doesn't show it at all.
3. For anything weak or missing, tell me if it's a real gap (I don't have the experience) or a resume gap (I have the experience but didn't show it) – these need different fixes.
4. For resume gaps, suggest specific language or bullet points to add using experience I actually have.
5. Give an honest overall assessment: strong match, worth applying with changes, or not a good fit yet.
```

CUSTOMIZE THESE

Paste the full job description, not just the title — the specific requirements listed are what get checked against, not the job title in general.

WHAT CLAUDE SHOULD GIVE YOU

A requirement-by-requirement match analysis, gaps split into “real” versus “resume-only,” suggested language for the fixable ones, and an honest fit verdict.

PRO TIP

For resume gaps, feed your rewritten bullets back into The Achievement Bullet Converter (Prompt #17) to make sure they're stated with real impact, not just present.

PROMPT #17

The Achievement Bullet Converter

What it does: Converts duty-based resume lines into specific, achievement-based bullets, without inventing numbers you don't have.

Best for: *Job seekers whose resume reads like a list of duties instead of results*

COPY & PASTE INTO CLAUDE

```
Act as a resume writer specializing in achievement-based bullets for [TARGET ROLE] resumes.

Current bullets:
[PASTE CURRENT BULLETS]

Rewrite each one using: Action verb + what you did + how + measurable result (where a real result exists).

Rules:
- Never invent a metric I didn't give you. If there's no real number, rewrite for specificity and impact without fabricating one, and flag it as needing a real metric.
- Cut generic openers like "responsible for" or "helped with."
- Keep each bullet to one line.

After the rewrite, tell me which bullets are now strong enough to use as-is, and which still need a specific piece of missing information from me (a number, a scope, a tool name) before they're truly strong.
```

CUSTOMIZE THESE

Include any real numbers you have (even rough ones — “about 30%”) in your original bullets so Claude can use them instead of leaving a gap.

WHAT CLAUDE SHOULD GIVE YOU

Rewritten bullets in the achievement format, an honest flag on any that still need a real metric from you, and a clear split between “ready to use” and “needs more input.”

PRO TIP

Do this per target role rather than once for a generic resume — the same experience should be framed differently depending on what the next job actually values.

PROMPT #18

The Portfolio Strategy Builder

What it does: Builds a project portfolio strategy based on a specific target role, so your projects actually demonstrate what that role is hiring for.

Best for: *Developers or creators deciding what projects to build or showcase for a specific target role*

COPY & PASTE INTO CLAUDE

```
Act as a career strategist advising on portfolio projects for [TARGET ROLE].

Target role: [TARGET ROLE, be specific – e.g. "junior backend developer, fintech"]
My current projects/skills: [LIST WHAT YOU HAVE]
Time available to build more: [TIMEFRAME]

Build a portfolio strategy:
1. Identify what a hiring manager for this specific role actually wants to see demonstrated – not generic "good projects," but the specific skills/signals this role screens for.
2. Evaluate my current projects against that: which ones already demonstrate the right things, which are irrelevant to this target, and which are close but missing something.
3. If I need 1-2 new projects, suggest specific project ideas that would fill the biggest gaps – scoped to fit my available time.
4. Tell me how to present the strongest existing project (what to highlight, what story to tell about it) to make its relevance obvious to a recruiter.
```

CUSTOMIZE THESE

Be as specific as possible about the target role — “frontend developer” and “frontend developer at an early-stage startup” get very different strategy advice.

WHAT CLAUDE SHOULD GIVE YOU

An honest audit of your current projects against the target role, gap-filling project ideas scoped to your real time budget, and presentation guidance for your strongest existing work.

PRO TIP

Once you pick a new project idea, run it through The MVP Scope Definer (Prompt #43) to keep it small enough to actually finish.

PROMPT #19

The Skill Gap Analyzer

What it does: Compares your current skills against a real job description and identifies the specific, prioritized gaps — not a generic “learn more” list.

Best for: *Job seekers targeting a specific role who need to know exactly what to learn next*

COPY & PASTE INTO CLAUDE

```
Act as a career advisor analyzing skill gaps for a specific job.

Job description:
[PASTE JOB DESCRIPTION]

My current skills/experience: [LIST YOUR SKILLS]
Experience level: [EXPERIENCE LEVEL]

Analyze the gap:
1. List the skills/requirements from the job description I already clearly have.
2. List the ones I'm missing or weak on, split into: critical (likely to disqualify me if unaddressed) and secondary (nice to have, learnable on the job).
3. For each critical gap, suggest the fastest realistic way to close it – or to convincingly demonstrate transferable experience if closing it fully isn't realistic before applying.
4. Give me an honest read: apply now and address gaps in the interview, apply after closing 1-2 specific gaps, or this role is a stretch right now.
```

CUSTOMIZE THESE

List your real current skills specifically, not a general resume summary — the gap analysis is only as accurate as what you tell it you already have.

WHAT CLAUDE SHOULD GIVE YOU

A skills-you-have list, a prioritized gap list split by how critical each one is, a fastest-path suggestion for critical gaps, and an honest apply-now-or-wait recommendation.

PRO TIP

For any gap you decide to close before applying, hand it directly to The Learning Path Architect for a structured plan instead of studying it unsystematically.

PROMPT #20

The Mock Technical Interviewer

What it does: Runs a realistic mock technical interview one question at a time, with feedback after each answer, ending in an honest readiness assessment.

Best for: *Job seekers who want realistic interview practice with honest feedback, not just a list of questions*

COPY & PASTE INTO CLAUDE

```
Act as an experienced technical interviewer for a [ROLE] position.

Interview focus: [TOPIC – e.g. data structures, system design, SQL, behavioral]
My level: [EXPERIENCE LEVEL]

Rules:
1. Ask one question at a time. Don't move on until I answer.
2. After each answer, give brief honest feedback – what was strong, what was missing or wrong – then ask the next question.
3. Vary difficulty realistically for this role; don't make every question maximally hard.
4. After [NUMBER] questions, give an overall assessment: readiness level, top 2 strengths, top 2 gaps to close before the real interview.

Start with your first question.
```

CUSTOMIZE THESE

Set a specific role, topic focus, and honest experience level before starting — these determine question difficulty far more than anything else.

WHAT CLAUDE SHOULD GIVE YOU

A realistic one-at-a-time interview with targeted feedback per answer, ending in a concrete readiness assessment naming your top strengths and gaps.

PRO TIP

Do the whole mock interview in a single conversation — Claude's feedback gets sharper as it sees more of your answering patterns across questions.

CATEGORY 4

Research

For turning scattered information into structured understanding — using Claude's ability to hold and reason across large amounts of material.

PROMPT #21

The Research Question Sharpener

What it does: Turns a vague topic into a set of sharp, answerable research questions before you spend time gathering sources.

Best for: *Anyone starting research on a vague topic who needs a focused starting point*

COPY & PASTE INTO CLAUDE

```
Act as a research advisor helping me scope a research project.

Topic I want to research: [TOPIC]
Why I'm researching this: [GOAL/PURPOSE]
What I already know: [CURRENT KNOWLEDGE, OR "NOT MUCH"]

Help me sharpen this:
1. Point out what's too vague or too broad about the topic as I've stated it.
2. Propose 4-6 specific, answerable research questions that would actually satisfy my stated goal
– avoid questions so broad they can't be answered with real sources.
3. Rank them by which would give me the most useful information first.
4. Flag any question that's likely to have limited or conflicting information available, so I know
to expect that going in.
```

CUSTOMIZE THESE

Be honest about your real purpose, not just the topic — the same topic needs very different research questions depending on whether you're writing a paper, making a decision, or just curious.

WHAT CLAUDE SHOULD GIVE YOU

A clear-eyed critique of your original framing, 4-6 sharp and rankable research questions, and a heads-up on which ones may have limited available information.

PRO TIP

Once you have your questions, come back with what you find and use The Multi-Source Research Synthesizer (Prompt #26) to pull it together.

PROMPT #22

The Technology Comparison Matrix Builder

What it does: Builds a structured, criteria-based comparison between technology options instead of a vague pros-and-cons list.

Best for: *Developers or teams choosing between tools, frameworks, or technologies*

COPY & PASTE INTO CLAUDE

```
Act as a technical advisor helping me choose between technologies.

Options I'm considering: [OPTION A] vs [OPTION B] (vs [OPTION C])
What I'm building: [PROJECT CONTEXT]
What matters most to me: [CRITERIA – e.g. learning curve, performance, ecosystem, long-term maintenance]

Build the comparison:
1. Compare the options specifically against my stated criteria, in priority order – not a generic feature list.
2. For each criterion, state which option wins and why, with enough specificity that I understand the actual tradeoff, not just the winner.
3. Flag anything about my specific project context that should weight the decision unusually (e.g. team size, timeline, existing stack).
4. Give a direct recommendation for my situation, and the one condition under which you'd recommend the opposite.
```

CUSTOMIZE THESE

State your real priorities, not a generic “which is better” — the same two technologies can have opposite “winners” depending on what you’re optimizing for.

WHAT CLAUDE SHOULD GIVE YOU

A criteria-ranked comparison with reasoning per criterion (not just a verdict), context-specific weighting notes, and a direct recommendation with its caveat.

PRO TIP

If the recommendation is close, ask Claude to describe the specific project scenario where the losing option would actually be the better choice — it clarifies what you’re really trading off.

PROMPT #23

The Argument Weakness Finder

What it does: Finds the actual weak points in an argument or piece of writing — logical gaps, unsupported claims, and the strongest counterargument — instead of general feedback.

Best for: Researchers, writers, or students who want to pressure-test an argument or paper before finalizing it

COPY & PASTE INTO CLAUDE

```
Act as a critical peer reviewer, not a supportive one.

Here's the argument/piece:

[PASTE ARGUMENT, ESSAY, OR PAPER SECTION]

Find the real weaknesses:
1. Identify any claims that aren't actually supported by the evidence given, even if they sound reasonable.
2. Point out logical gaps – places where a conclusion doesn't fully follow from what came before it.
3. Identify the strongest counterargument someone could make against this, and how well (or poorly) the piece currently addresses it.
4. Flag anywhere the argument relies on an unstated assumption that a skeptical reader wouldn't automatically grant.

Don't soften this to be encouraging. I want the version of this feedback that actually makes the argument stronger.
```

CUSTOMIZE THESE

Paste the actual argument text, not a summary of it — the specific wording is what reveals unsupported claims and logical gaps.

WHAT CLAUDE SHOULD GIVE YOU

A direct list of unsupported claims, logical gaps, the strongest counterargument with an honest read on how well it's currently handled, and any unstated assumptions.

PRO TIP

After revising based on the feedback, run the new version through the same prompt again — fixing one weakness sometimes introduces or exposes another.

PROMPT #24

The Long-Document Actionable Extractor

What it does: Extracts only the information from a long document that's actually actionable for your specific purpose — skipping context you don't need.

Best for: *Anyone who needs to act on a long document (report, contract excerpt, research paper, transcript) without reading all of it*

COPY & PASTE INTO CLAUDE

```
Act as an analyst extracting only what's useful from a long document.

Document:
[PASTE FULL DOCUMENT OR LONG TEXT]

Why I'm reading this / what I need to do with it: [PURPOSE]

Extract only what serves that purpose:
1. Anything that requires a decision or action from me, stated plainly.
2. Any deadline, number, or commitment that matters given my purpose.
3. Anything that contradicts what I'd normally assume, or that seems like it could cause a problem if missed.
4. Skip background, boilerplate, and anything not relevant to my stated purpose — don't summarize the whole document, extract from it.

End with: based on this document, what's the one thing I most need to do next?
```

CUSTOMIZE THESE

State your real purpose specifically — the same document yields a completely different extraction depending on whether you're deciding, reporting, or just need to flag risks.

WHAT CLAUDE SHOULD GIVE YOU

A purpose-filtered extraction (not a full summary): action items, key numbers/deadlines, anything contradicting expectations, and one clear next step — with irrelevant background left out.

PRO TIP

For very long documents, paste it in sections and ask Claude to hold a running list of extracted items across your messages instead of restarting each time.

PROMPT #25

The Messy Research to Structured Notes Converter

What it does: Turns scattered research material — quotes, links, half-formed notes — into a structured reference organized by theme, with sources preserved.

Best for: Researchers with scattered notes, quotes, and links from multiple sources who need it organized

COPY & PASTE INTO CLAUDE

```
Act as a research assistant organizing scattered material.

Here's everything I've gathered on [TOPIC]:

[PASTE MESSY NOTES, QUOTES, LINKS, FRAGMENTS]

Organize this into a structured reference:
1. Group the material into themes/subtopics (not just the order I collected it in).
2. Under each theme, summarize what the material says, keeping track of which source each point came from.
3. Flag any direct contradictions between sources.
4. Flag any theme where I clearly need more information – where my current material is thin.
5. End with a short list of the most important open questions this material doesn't yet answer.
```

CUSTOMIZE THESE

Paste the material as-is, including messy fragments and unlabeled quotes — don't pre-organize it, that's the part Claude is doing for you.

WHAT CLAUDE SHOULD GIVE YOU

A theme-organized reference with sources attributed per point, flagged contradictions between sources, thin areas needing more research, and a list of remaining open questions.

PRO TIP

Feed the flagged “thin” areas back into The Research Question Sharpener (Prompt #21) to figure out exactly what to go find next.

PROMPT #26

The Multi-Source Research Synthesizer

What it does: Synthesizes findings from multiple sources into one coherent brief, explicitly noting where sources agree, disagree, or leave gaps.

Best for: *Anyone who has gathered information from several sources and needs one coherent picture*

COPY & PASTE INTO CLAUDE

```
Act as a research analyst synthesizing multiple sources into a brief.

Sources/findings:

Source 1: [PASTE OR SUMMARIZE]
Source 2: [PASTE OR SUMMARIZE]
Source 3: [PASTE OR SUMMARIZE]
(add more as needed)

My purpose: [WHAT YOU NEED THIS BRIEF FOR]

Synthesize this:
1. Write a short overview of the consensus view across sources, where one exists.
2. Explicitly call out where sources disagree, and what might explain the disagreement (different methodology, timeframe, definitions, bias).
3. Identify what none of the sources address, but that matters for my stated purpose.
4. Give a final synthesized take: given everything above, what's the most defensible conclusion right now?

Keep the brief focused on my purpose — don't just restate each source in sequence.
```

CUSTOMIZE THESE

Include enough of each source's actual content (not just a title) so Claude can identify real agreement and disagreement, not just topic overlap.

WHAT CLAUDE SHOULD GIVE YOU

A synthesized brief with a consensus overview, explicitly flagged disagreements with possible explanations, identified gaps relative to your purpose, and one defensible conclusion.

PRO TIP

If sources disagree significantly, ask Claude to lay out what evidence would resolve the disagreement — useful for deciding whether it's worth researching further.

Productivity

For getting unstuck, deciding what actually matters, and catching problems in a plan before they cost you time.

PROMPT #27

The Overwhelming Task Breakdown Tool

What it does: Breaks an overwhelming task down into a concrete first action and a realistic sequence, so “too big to start” becomes a clear next step.

Best for: *Anyone staring at a task that feels too big to start*

COPY & PASTE INTO CLAUDE

```
Act as a productivity coach who specializes in breaking down overwhelming tasks.

The task I'm stuck on: [TASK]
Why it feels overwhelming: [WHAT'S MAKING IT HARD – e.g. too many unknowns, too big, don't know where to start]
Deadline (if any): [DEADLINE]

Break it down:
1. Identify why this specific task feels overwhelming – is it actually one big task, or several smaller tasks disguised as one?
2. Break it into a sequence of concrete, individually completable steps.
3. Identify the smallest possible first step I could do in the next 15 minutes to build momentum.
4. Flag any step that's actually a decision I'm avoiding, rather than a task – name the decision directly.
```

CUSTOMIZE THESE

Be honest about why it feels overwhelming — that diagnosis (too big vs. too vague vs. avoiding a decision) changes what breakdown actually helps.

WHAT CLAUDE SHOULD GIVE YOU

A named reason for the overwhelm, a concrete step sequence, one genuinely tiny first action, and any hidden decisions the task list was disguising as work.

PRO TIP

Actually do the 15-minute first step before doing anything else with the plan — momentum from a completed tiny step matters more than a perfect breakdown.

PROMPT #28

The Weekly Priority Planner

What it does: Turns a messy task list into a prioritized, realistic weekly plan based on what actually matters, not just what feels urgent.

Best for: *Anyone with more on their plate than hours in the week*

COPY & PASTE INTO CLAUDE

```
Act as a productivity coach who prioritizes by impact, not urgency.

Everything on my plate this week: [LIST/PASTE TASKS, MESSY IS FINE]
My biggest goal this week: [GOAL]
Hours realistically available: [HOURS]
Hard deadlines: [DEADLINES, IF ANY]

Build the plan:
1. Sort into: Must do this week / Should do if time allows / Can wait.
2. Identify the 1-3 tasks most tied to my biggest goal, even if others feel more urgent.
3. Flag anything that looks unnecessary, delegable, or reducible – and say why.
4. Give a rough day-by-day distribution based on my real available hours.
5. If my list is unrealistic for the time I have, say so directly and tell me what to cut.
```

CUSTOMIZE THESE

Dump the real, messy list without pre-sorting it yourself — part of the value is Claude catching what you'd naturally over-prioritize just because it feels urgent.

WHAT CLAUDE SHOULD GIVE YOU

A sorted task list, the goal-critical 1-3 tasks, honest cut/delegate flags, a day-by-day distribution, and a direct reality check if you've overloaded the week.

PRO TIP

Run this at the start of every week with the same phrasing — consistency makes it easy to notice which tasks keep landing in “can wait” and should probably be dropped entirely.

PROMPT #29

The Decision Pressure Test

What it does: Forces a real tradeoff analysis on a decision you're stuck on, instead of a pros-and-cons list that treats every factor as equally weighted.

Best for: *Anyone stuck between options and wanting the tradeoffs made explicit before choosing*

COPY & PASTE INTO CLAUDE

Act as an advisor helping me make a decision by pressure-testing it, not just listing pros and cons.

Decision: [WHAT YOU'RE DECIDING BETWEEN]

Options: [OPTION A] vs [OPTION B] (vs more)

What matters most to me, in priority order: [YOUR PRIORITIES]

Constraints: [CONSTRAINTS – time, money, reversibility, etc.]

Pressure-test this:

1. For each option, state the real cost – not just the downside, but what you're giving up by choosing it.
2. Weigh the options specifically against my stated priorities, in the order I gave them – don't treat all factors as equal.
3. Identify which option is more reversible if it turns out wrong, and whether that should matter here.
4. Tell me what additional piece of information, if I had it, would make this decision obvious – and whether it's worth delaying to go get it.
5. Give a direct recommendation given everything above, not just a summary of both sides.

CUSTOMIZE THESE

Rank your actual priorities, don't just list them — the forced weighting is what makes the tradeoff analysis useful instead of a neutral pros-and-cons list.

WHAT CLAUDE SHOULD GIVE YOU

A real-cost breakdown per option weighted by your stated priorities, a reversibility assessment, the key missing information (if any) worth waiting for, and a direct recommendation.

PRO TIP

If you disagree with the recommendation, say so and ask why — the disagreement itself often reveals a priority you didn't actually state up front.

PROMPT #30

The Goal-to-Action-Plan Converter

What it does: Converts a stated goal into a concrete action plan with milestones, instead of leaving it as an intention.

Best for: *Anyone with a goal but no concrete plan to actually get there*

COPY & PASTE INTO CLAUDE

```
Act as a strategist who turns goals into executable plans.

Goal: [GOAL]
Current situation: [WHERE YOU'RE STARTING FROM]
Timeframe: [TIMEFRAME]
Resources/constraints: [WHAT YOU HAVE OR DON'T HAVE – time, money, skills, support]

Build the plan:
1. Break the goal into 3-5 concrete milestones between now and the deadline.
2. For the first milestone, list the specific actions needed – specific enough to put directly on a calendar.
3. Identify the single biggest risk to this plan failing, and build in a mitigation for it.
4. Tell me how I'll know if I'm on track at the halfway point, not just at the end.
5. If the timeframe is unrealistic given my resources, say so and suggest an adjusted version.
```

CUSTOMIZE THESE

Be specific about your real starting point and real constraints — a generic “I want to learn X” goal produces a generic plan; a stated starting point and constraint produces an executable one.

WHAT CLAUDE SHOULD GIVE YOU

A milestone-based plan, a concrete first-milestone action list, one named biggest risk with a mitigation, a midpoint check-in criterion, and an honest timeline reality check.

PRO TIP

Come back at the halfway checkpoint and ask Claude to evaluate your actual progress against the plan — adjust before the deadline, not at it.

PROMPT #31

The Project Pre-Mortem

What it does: Runs a pre-mortem on a project — imagining it has already failed and working backward to find out why, so you can prevent it now instead of diagnosing it later.

Best for: *Anyone starting a project who wants to catch failure points before they happen, not after*

COPY & PASTE INTO CLAUDE

```
Act as a strategist running a pre-mortem on my project.

Project: [DESCRIBE PROJECT]
Goal/success criteria: [WHAT SUCCESS LOOKS LIKE]
Timeline: [TIMELINE]

Run the pre-mortem:
1. Imagine it's [TIMELINE] from now and this project has clearly failed. Generate the 5 most plausible reasons why, based on what you know about projects like this.
2. For each reason, rate how likely it is and how much warning I'd realistically get before it became a real problem.
3. For the top 2-3 most likely/most damaging risks, suggest a specific preventive action I could take now, not after the fact.
4. Identify one assumption I'm probably making that, if wrong, would quietly undermine the whole project.
```

CUSTOMIZE THESE

Describe the project and its real success criteria specifically — vague projects produce generic failure modes instead of the ones actually likely to hit you.

WHAT CLAUDE SHOULD GIVE YOU

Five plausible failure scenarios ranked by likelihood and warning time, concrete preventive actions for the top risks, and one dangerous unstated assumption worth checking now.

PRO TIP

Revisit this pre-mortem at your project's midpoint with what's actually happened so far — it's far more useful as a recurring check than a one-time exercise.

PROMPT #32

The Assumption Challenger

What it does: Deliberately challenges your assumptions about a plan or belief instead of validating them, to surface blind spots before they cost you.

Best for: *Anyone who wants their plan or belief pressure-tested instead of agreed with*

COPY & PASTE INTO CLAUDE

Act as a sharp, skeptical advisor whose job is to challenge my thinking, not agree with it.

Here's my plan/belief: [DESCRIBE YOUR PLAN OR BELIEF]

Why I think this is right: [YOUR REASONING]

Challenge it:

1. Identify the assumptions I'm making that I haven't actually verified – things I'm treating as fact that are really guesses.
2. Make the strongest possible case against my plan/belief, as if you were trying to talk someone out of it.
3. Point out anywhere my reasoning has a gap or relies on wishful thinking rather than evidence.
4. If, after all that, my plan still holds up – say so honestly. Don't manufacture disagreement for its own sake, but don't soften real problems either.

CUSTOMIZE THESE

State your actual reasoning, not just the conclusion — the challenge is only useful if it's attacking the real logic you're relying on, not a guess at it.

WHAT CLAUDE SHOULD GIVE YOU

A list of unverified assumptions, the strongest honest counter-case to your plan, flagged reasoning gaps, and a genuinely honest verdict — including “this actually holds up” if that's true.

PRO TIP

Use this before committing real time or money to a plan, not after — it's designed to be uncomfortable early, when it's still cheap to change course.

Writing & Content

For sharpening writing, finding the real hook, and turning raw material into something worth publishing.

PROMPT #33

The Unclear Writing Rewriter

What it does: Diagnoses exactly what's making a piece of writing unclear, then rewrites it — so you learn what was wrong, not just get a replacement.

Best for: *Anyone whose writing feels muddled but can't pinpoint why*

COPY & PASTE INTO CLAUDE

```
Act as an editor specializing in clarity.

Here's what I wrote:

[PASTE TEXT]

Who this is for: [AUDIENCE]
What I want them to understand or do after reading it: [GOAL]

Do this in order:
1. Identify specifically what's making this unclear – buried point, too much jargon, sentences doing too much at once, missing structure, etc. Name the actual problem, don't just say "it's unclear."
2. Rewrite it for clarity, keeping my original voice and key points intact.
3. Show me the single most important sentence in the piece, and confirm it's positioned where a reader won't miss it.
4. Point out one pattern in my original writing that's worth fixing going forward, not just in this piece.
```

CUSTOMIZE THESE

Paste your actual draft and be specific about the audience and what you want them to do — “clear” means different things depending on who’s reading and why.

WHAT CLAUDE SHOULD GIVE YOU

A named diagnosis of what was actually unclear, a rewrite that preserves your voice, confirmation your key point is positioned well, and one recurring pattern to watch for in future writing.

PRO TIP

Ask specifically “what’s a pattern I do across multiple pieces?” after doing this on 2-3 different drafts — individual fixes are useful, but the recurring pattern is what actually improves your writing.

PROMPT #34

The Hook Improver

What it does: Diagnoses why an opening line or hook isn't working and generates stronger alternatives grounded in what actually made the original weak.

Best for: Content creators and writers whose opening lines aren't earning attention

COPY & PASTE INTO CLAUDE

```
Act as an editor who specializes in opening hooks.

Current opening: [PASTE YOUR OPENING LINE/PARAGRAPH]
What this piece is about: [TOPIC]
Platform/format: [PLATFORM, e.g. blog post, Instagram caption, video script]
Target audience: [AUDIENCE]

Improve it:
1. Diagnose specifically why the current opening isn't earning attention – too generic, buries the interesting part, doesn't create curiosity, etc.
2. Generate 5 alternative hooks using different angles: a surprising fact/claim, a specific pain point, a curiosity gap, a bold/contrarian statement, and a before/after framing.
3. For each, note the tradeoff – what kind of attention it earns and what it risks (e.g. clickbait feel, requires a strong payoff).
4. Recommend which one best fits my actual content and audience, not just which sounds punchiest.
```

CUSTOMIZE THESE

Give the real topic and platform — a hook that works for a blog post often fails on a 3-second Reel, and vice versa.

WHAT CLAUDE SHOULD GIVE YOU

A specific diagnosis of why the current hook underperforms, five differently-angled alternatives with their tradeoffs named, and a recommendation matched to your actual content.

PRO TIP

Make sure the piece actually delivers on whatever the winning hook promises — a strong hook with a weak payoff hurts trust more than a mediocre hook would.

PROMPT #35

The Notes-to-Polished-Content Converter

What it does: Turns rough notes or bullet points into a polished, structured piece of content in your intended format — without smoothing your voice into something generic.

Best for: *Creators and writers who have rough ideas or bullet points but not a finished piece*

COPY & PASTE INTO CLAUDE

```
Act as a writer/editor turning rough notes into finished content.

My notes/bullet points: [PASTE NOTES]
Format: [FORMAT, e.g. blog post, LinkedIn post, email, video script]
Tone: [TONE, e.g. direct and practical, conversational, professional]
Audience: [AUDIENCE]

Turn this into a finished piece:
1. Identify the strongest single idea in my notes and structure the piece around that as the throughline — don't just stitch every note together in order.
2. Write the full piece in the specified format and tone.
3. Keep language and phrasing close to how I'd actually say it based on my notes — don't over-polish into generic-sounding writing.
4. Flag anything from my notes you left out, and why.
```

CUSTOMIZE THESE

Paste real notes, even fragmented ones — and be specific about tone, since “professional” and “conversational” produce very different pieces from the same notes.

WHAT CLAUDE SHOULD GIVE YOU

A finished piece structured around your strongest idea (not just your notes in order), written in your specified tone, plus a transparent note on anything cut and why.

PRO TIP

If it sounds too generic, paste back a piece of your past writing and ask Claude to match that voice specifically instead of the tone description alone.

PROMPT #36

The Content Repurposing Engine

What it does: Turns one piece of long-form content into multiple platform-specific formats, adapted to how each platform is actually consumed, not just resized.

Best for: *Creators who write one piece of long-form content and want it working across platforms*

COPY & PASTE INTO CLAUDE

```
Act as a content strategist who repurposes long-form content across platforms.

Original content:
[PASTE FULL BLOG POST / VIDEO SCRIPT / ARTICLE]

Repurpose this into:
1. A short-form hook + caption for [PLATFORM, e.g. Instagram/TikTok] that captures the single most shareable idea, not a summary of everything.
2. A LinkedIn post version, adapted for a more professional, insight-driven tone.
3. A 5-tweet/thread version that stands on its own without needing the original.
4. One "quote card" worthy line pulled directly from the content.

For each format, adapt the structure to how people actually consume that platform — don't just shorten the original.
```

CUSTOMIZE THESE

Paste the full original content, not a summary — repurposing works best when Claude can pull specific lines and ideas directly from the source material.

WHAT CLAUDE SHOULD GIVE YOU

Platform-adapted versions (not just shortened copies) for short-form video/caption, LinkedIn, a thread, and a standalone quote line — each suited to how that platform is actually read.

PRO TIP

Ask which single idea from the original has the most “standalone” potential before repurposing — that becomes your anchor and keeps every version focused instead of diluted.

PROMPT #37

The Audience Needs Analyzer

What it does: Analyzes what a specific audience actually needs and wants from your content, based on their real situation, not generic content-strategy advice.

Best for: *Creators and marketers unsure what their audience actually wants from their content*

COPY & PASTE INTO CLAUDE

```
Act as an audience strategist.

My audience: [DESCRIBE YOUR AUDIENCE – who they are, what they're trying to do]
What I currently create content about: [YOUR TOPIC/NICHE]
A recent piece I made: [PASTE OR DESCRIBE IT]

Analyze the fit:
1. Based on my audience's real situation, what are they actually trying to figure out or accomplish that I could help with?
2. Evaluate my recent piece against that – does it address a real need, or is it interesting-to-me but not clearly useful-to-them?
3. Identify one gap between what I'm currently creating and what my audience most needs.
4. Suggest 3 content angles that would close that gap, specific enough to start outlining immediately.
```

CUSTOMIZE THESE

Describe your audience by situation and goal, not just demographics — “beginner developers trying to get their first job” is far more useful than “people interested in coding.”

WHAT CLAUDE SHOULD GIVE YOU

A grounded read on what your audience actually needs, an honest evaluation of a recent piece against that need, one named content gap, and 3 specific, outline-ready content angles.

PRO TIP

Run this quarterly, not once — what your audience needs shifts as they grow past their earlier problems, and content strategy should shift with it.

PROMPT #38

The Outline Architect

What it does: Builds a content outline structured around a single throughline and a clear reader payoff, instead of a flat list of subtopics.

Best for: *Writers who want a strong structure before drafting, not just a topic-to-topic list*

COPY & PASTE INTO CLAUDE

```
Act as a content strategist building an outline.
```

```
Topic: [TOPIC]
```

```
Format: [FORMAT, e.g. blog post, video, guide]
```

```
Audience: [AUDIENCE]
```

```
What the reader should walk away with: [OUTCOME]
```

```
Build the outline:
```

1. State the single throughline this piece should build toward — the one idea everything else should serve.
2. Structure sections so each one builds on the last, not just a list of related subtopics in arbitrary order.
3. For each section, give a one-line brief on the specific point it needs to make (not just a topic label).
4. Identify where the piece is most likely to lose the reader's attention, and suggest how to handle that section to keep momentum.

CUSTOMIZE THESE

State the intended reader outcome specifically — “understand X” and “be able to do X” produce structurally different outlines.

WHAT CLAUDE SHOULD GIVE YOU

An outline built around one clear throughline, sections that build on each other with a one-line brief each, and a flagged attention-risk point with a fix.

PRO TIP

Draft the section most likely to lose attention first, while you have the most energy for it, instead of writing linearly from the introduction.

CATEGORY 7

Business & Entrepreneurship

For stress-testing ideas, understanding customers and competitors, and scoping what to actually build first.

PROMPT #39

The Idea Weakness Finder

What it does: Finds the specific weaknesses in a business or product idea, instead of general encouragement or generic startup advice.

Best for: Founders who want an honest gut-check on an idea before investing real time

COPY & PASTE INTO CLAUDE

```
Act as a skeptical but fair advisor who has seen many early-stage ideas succeed and fail.

My idea: [DESCRIBE IDEA IN 2-4 SENTENCES]
Target customer: [WHO IT'S FOR]
Problem it solves: [PROBLEM]

Find the real weaknesses:
1. Is this solving a problem people actually feel and would pay to fix, or a mild inconvenience they'll tolerate?
2. Who already solves this, directly or indirectly – and why would someone actually switch to me?
3. What's the biggest reason this specific idea could fail, given what you know about ideas like it?
4. What's the fastest, cheapest way I could get real evidence about whether I'm wrong, before building anything?

Don't soften this to be encouraging. If it's weak, say so and explain exactly why.
```

CUSTOMIZE THESE

Describe the idea the way you'd explain it to a stranger, not the way you'd pitch it — the critique is only as sharp as what you actually give it to work with.

WHAT CLAUDE SHOULD GIVE YOU

A direct assessment of problem strength, existing competition and switching cost, the most likely failure mode, and one concrete low-cost way to test it before building.

PRO TIP

Ask for the strongest honest case FOR the idea afterward, too — seeing both sides argued well is more useful than either alone for actually deciding.

PROMPT #40

The Competitor Analysis Builder

What it does: Builds a structured competitor analysis focused on where real openings exist, not just a feature-by-feature comparison table.

Best for: *Founders and marketers who need to understand the competitive landscape before positioning their product*

COPY & PASTE INTO CLAUDE

```
Act as a competitive strategist.

My product/idea: [DESCRIBE IT]
Competitors I know of: [LIST COMPETITORS]
Target customer: [WHO IT'S FOR]

Build the analysis:
1. For each competitor, identify what they do well and what they're likely weak on, based on what you know.
2. Identify the gap – what does my target customer need that none of these competitors are serving well right now?
3. Assess how defensible that gap actually is – is it a real opening, or something a competitor could copy easily once I prove it works?
4. Recommend a specific positioning angle based on this analysis, not a generic "better, faster, cheaper."
```

CUSTOMIZE THESE

List real, specific competitors (not just a category) — the analysis depends on Claude reasoning about actual products, not a generic competitive landscape.

WHAT CLAUDE SHOULD GIVE YOU

A per-competitor strengths/weaknesses read, an identified customer gap, an honest defensibility assessment of that gap, and a specific positioning recommendation.

PRO TIP

Ask “what would [STRONGEST COMPETITOR] do if they saw me succeeding with this positioning?” — it pressure-tests whether your gap is actually defensible.

PROMPT #41

The Target User Definer

What it does: Narrows a vague target user description into a specific, usable definition — sharp enough to actually guide product and marketing decisions.

Best for: Founders whose “target user” is currently too broad to actually design or market for

COPY & PASTE INTO CLAUDE

```
Act as a product strategist helping me define my real target user.

My product/idea: [DESCRIBE IT]
Who I currently think my target user is: [YOUR CURRENT DESCRIPTION]

Sharpen this:
1. Point out specifically why my current definition is too broad or too vague to actually design or market for.
2. Propose 2-3 more specific user segments within my current broad definition, each with a distinct situation and need.
3. For each segment, identify what would make my product a clear yes for them versus a maybe.
4. Recommend which segment to focus on first, and why – based on urgency of their need and how reachable they are, not just market size.
```

CUSTOMIZE THESE

Give your current honest (even if vague) definition — the value is in Claude showing you specifically where it's too broad, which only works against a real starting point.

WHAT CLAUDE SHOULD GIVE YOU

A specific critique of your current definition, 2-3 sharper user segments with distinct needs, clarity on what makes each a clear yes, and a reasoned recommendation on which to focus first.

PRO TIP

Once you pick a segment, run The Customer Pain Point Extractor (Prompt #42) specifically for that narrower group instead of your original broad audience.

PROMPT #42

The Customer Pain Point Extractor

What it does: Extracts the real underlying pain points from customer feedback, reviews, or conversation notes — separating surface complaints from the deeper problem.

Best for: *Founders who need to understand what customers actually struggle with, from raw feedback or conversations*

COPY & PASTE INTO CLAUDE

```
Act as a product researcher analyzing customer feedback.

Raw feedback/notes:
[PASTE REVIEWS, SURVEY RESPONSES, OR CONVERSATION NOTES]

Extract the real pain points:
1. Group the feedback into themes, not individual quotes.
2. For each theme, distinguish between the surface complaint (what they literally said) and the underlying pain point (what's actually frustrating them).
3. Rank the themes by how frequently they appear and how strongly they seem to be felt.
4. Flag anything that looks like a request for a specific feature versus a signal of a deeper unmet need — these should be treated differently.
```

CUSTOMIZE THESE

Paste real feedback text, not your summary of it — the surface-complaint-versus-underlying-need distinction depends on the actual wording people used.

WHAT CLAUDE SHOULD GIVE YOU

Theme-grouped feedback with surface complaint versus real underlying pain point separated per theme, a frequency/intensity ranking, and flagged feature-requests-versus-deeper-needs.

PRO TIP

For your top-ranked pain point, ask Claude to draft 3 clarifying questions you could ask customers directly to confirm you've correctly identified the underlying need.

PROMPT #43

The MVP Scope Definer

What it does: Defines the actual minimum viable version of a product idea — cutting scope down to what's needed to test the core assumption, not to launch a full product.

Best for: *Founders and builders who tend to over-scope their first version*

COPY & PASTE INTO CLAUDE

```
Act as a product advisor who specializes in ruthless MVP scoping.

Product idea: [DESCRIBE FULL IDEA/VISION]
The core assumption I need to test: [WHAT YOU MOST NEED TO LEARN – e.g. will people pay for this / will they use it regularly]
Time/resources available: [CONSTRAINTS]

Scope the MVP:
1. Identify the single core assumption this MVP needs to test – if I only confirm this and nothing else, was it worth building?
2. List everything in my full vision that is NOT needed to test that assumption, and should be cut for now.
3. Define the smallest version that could still realistically test the core assumption, given my real constraints.
4. Flag anything I might be tempted to add "just in case" that would slow this down without adding to what I'm actually trying to learn.
```

CUSTOMIZE THESE

Name your core assumption honestly, even if it's uncomfortable — an MVP scoped around the wrong assumption tests the wrong thing no matter how minimal it is.

WHAT CLAUDE SHOULD GIVE YOU

A confirmed core assumption to test, an explicit cut list from your full vision, a scoped-down MVP definition that fits your real constraints, and a warning list of tempting-but-unnecessary additions.

PRO TIP

Once you ship the MVP, come back and describe what actually happened — use it to re-scope the next version instead of building out your original full vision by default.

PROMPT #44

The Business Model Stress Test

What it does: Pressure-tests a business model's core mechanics — how it actually makes money and what has to be true for that to keep working — instead of just describing it back to you.

Best for: *Founders who want to check whether their business model actually holds up before scaling it*

COPY & PASTE INTO CLAUDE

```
Act as a business strategist stress-testing a business model.

My business model: [DESCRIBE HOW YOU MAKE MONEY – pricing, customer acquisition, what you deliver]
Stage: [STAGE, e.g. pre-launch, early customers, scaling]

Stress-test it:
1. State clearly what has to be true for this model to work at scale (customer acquisition cost, retention, margins, etc.) – even if I haven't given you exact numbers, flag what assumption I'm implicitly making.
2. Identify the single point in this model most likely to break first as it grows (e.g. acquisition gets expensive, support costs scale faster than revenue).
3. Compare this model to how similar businesses typically make money – is mine unusually fragile or fairly standard for this type of business?
4. Suggest one metric I should be tracking closely right now to catch a problem with this model early.
```

CUSTOMIZE THESE

Describe the actual mechanics of how you make money, not just the product — the stress test targets the business model specifically, separate from whether the product itself is good.

WHAT CLAUDE SHOULD GIVE YOU

The implicit assumptions your model depends on, the most likely breaking point as you scale, a comparison to how similar businesses typically work, and one metric worth tracking closely now.

PRO TIP

Come back to this stress test at each major growth stage — the breaking point that matters at 10 customers is rarely the one that matters at 1,000.

AI Workflows

For using Claude as an ongoing thinking partner, not just a one-off question-answerer.

PROMPT #45

The Thinking Partner Setup

What it does: Sets up Claude as an ongoing thinking partner for a specific project, with a defined role and standing instructions, instead of restarting context every message.

Best for: *Anyone working on an ongoing project who wants Claude to function as a consistent collaborator across a conversation*

COPY & PASTE INTO CLAUDE

```
Act as my thinking partner for [PROJECT/GOAL] for the rest of this conversation.

Context on the project: [DESCRIBE PROJECT/GOAL]
My role in it: [YOUR ROLE]
What I most need from you: [e.g. "push back on weak reasoning," "help me think through tradeoffs,"
"keep me honest about scope"]

Standing instructions for this conversation:
1. Don't just agree with my ideas – tell me directly if something seems weak, and why.
2. When I bring you a new problem related to this project, ask clarifying questions before jumping
to a solution if the problem is underspecified.
3. Keep track of decisions I've already made in this conversation and don't re-litigate them
unless I bring them up again.
4. If I seem to be avoiding a hard decision, name it directly.

Confirm you understand this role, then I'll bring you the first thing I'm working through.
```

CUSTOMIZE THESE

Be specific about what you actually want from the role — “push back” and “help me think through options neutrally” are different jobs, and the setup should match what you need.

WHAT CLAUDE SHOULD GIVE YOU

A confirmed understanding of its ongoing role, followed by consistent behavior across the conversation — pushback where warranted, clarifying questions on vague asks, and named avoidance if it shows up.

PRO TIP

Keep this project in one long conversation instead of starting fresh each time — the standing instructions and accumulated context are what make this different from a one-off prompt.

PROMPT #46

The Document Analysis Assistant

What it does: Analyzes a long document (contract, report, research paper, transcript) for exactly the kind of insight you need, instead of a generic summary.

Best for: *Anyone who needs to deeply analyze a long or complex document for specific insights*

COPY & PASTE INTO CLAUDE

```
Act as an analyst reviewing a document for a specific purpose.

Document:
[PASTE OR ATTACH DOCUMENT]

What I need from this analysis: [SPECIFIC PURPOSE – e.g. "find every commitment we're making,"
"assess the strength of this argument," "find what's missing compared to X"]

Analyze it for that specific purpose:
1. Go through the document with only my stated purpose in mind – don't produce a general summary.
2. Quote or reference the specific sections that matter most for my purpose.
3. Flag anything ambiguous, missing, or contradictory relevant to what I'm trying to determine.
4. Give a direct answer to what I actually need to know, not just a restatement of the document's
contents.
```

CUSTOMIZE THESE

State your specific purpose, not “analyze this” — the same document produces a completely different (and far more useful) analysis depending on what you're actually trying to find out.

WHAT CLAUDE SHOULD GIVE YOU

A purpose-driven analysis referencing specific sections, flagged ambiguities or gaps relevant to your need, and a direct answer to your actual question — not a generic summary.

PRO TIP

For very long or important documents, ask Claude to note its confidence level on any conclusion that required interpretation versus one stated explicitly in the text.

PROMPT #47

The Vague Idea to Execution Plan Converter

What it does: Takes a vague, unformed idea and turns it into a concrete execution plan by first forcing the vague parts into specifics.

Best for: *Anyone with a rough idea they haven't turned into anything concrete yet*

COPY & PASTE INTO CLAUDE

```
Act as a strategist who turns vague ideas into concrete plans.

My vague idea: [DESCRIBE IT, ROUGH IS FINE]

Turn this into something executable:
1. Identify what's actually vague about this idea right now – the specific unknowns standing between me and being able to start.
2. Ask me up to 4 clarifying questions that would resolve the biggest unknowns, if you genuinely need the answers before proceeding.
3. Once clarified (or using reasonable assumptions if I can't answer everything), turn this into a concrete first version: what it actually is, who it's for, and what the first tangible output would be.
4. Give me the single next action I could take today to start turning this from an idea into something real.
```

CUSTOMIZE THESE

Answer the clarifying questions honestly, even with a guess, rather than skipping them — the concreteness of the final plan depends on resolving the actual unknowns, not working around them.

WHAT CLAUDE SHOULD GIVE YOU

A named list of what's currently vague, targeted clarifying questions, a concretized first version of the idea, and one specific action you can take today.

PRO TIP

Once you have the concrete version, run it through The MVP Scope Definer (Prompt #43) if it's a product, or The Goal-to-Action-Plan Converter (Prompt #30) if it's a personal goal.

PROMPT #48

The Repeatable Workflow Builder

What it does: Turns a task you do repeatedly into a reusable prompt template you can save and reuse, instead of re-explaining context every time.

Best for: *Anyone doing the same kind of task regularly who wants a reusable system instead of starting from scratch each time*

COPY & PASTE INTO CLAUDE

```
Act as a workflow designer helping me build a reusable prompt.

The task I do repeatedly: [DESCRIBE THE RECURRING TASK]
What changes each time I do it: [VARIABLES – e.g. the specific topic, the specific code, the specific person]
What stays the same each time: [YOUR CONSISTENT GOAL/STANDARD/FORMAT]

Build me a reusable prompt template:
1. Identify what should be a fixed instruction (the consistent part) versus a placeholder (the part that changes each time).
2. Write the template with clear [PLACEHOLDERS] for the variable parts.
3. Include any quality standard or output format I've described as a fixed rule in the template, so I don't have to restate it every time.
4. Give me a short usage note on what to fill in and what NOT to change, so this stays reliable each time I reuse it.
```

CUSTOMIZE THESE

Describe several past instances of the task, not just one — seeing what actually varies versus what's always the same is what makes the placeholders accurate.

WHAT CLAUDE SHOULD GIVE YOU

A ready-to-save prompt template with clear placeholders, your consistent standards baked in as fixed instructions, and a short usage note for future you.

PRO TIP

Save the finished template somewhere you'll actually find it again (a notes app, a doc) — the whole point is skipping the re-explaining next time.

PROMPT #49

The Personal AI Assistant Config

What it does: Builds a standing “configuration” prompt that sets up Claude to reliably act in a specific role — like a weekly reviewer or accountability partner — every time you start that kind of conversation.

Best for: *Anyone who wants Claude to consistently act a certain way for a specific ongoing role in their life or work*

COPY & PASTE INTO CLAUDE

```
Act as my [ROLE, e.g. weekly accountability partner / writing editor / study coach] going forward.

Context you should know: [RELEVANT BACKGROUND – your goals, standards, current situation]
How I want you to behave in this role: [BEHAVIOR RULES – e.g. "always ask what I actually did
before accepting my plan for next week," "never soften critical feedback"]
What this role is NOT responsible for: [BOUNDARIES, IF ANY]

Confirm you understand this role and these rules. Then, each time I start a conversation with you
acting as my [ROLE], I'll begin by saying "[ROLE] session" and giving you the update – respond
according to the rules above from that point on.
```

CUSTOMIZE THESE

Write behavior rules as specific, checkable instructions (“always ask X before Y”) rather than vague tone descriptions (“be helpful”) — specific rules are what actually get followed consistently.

WHAT CLAUDE SHOULD GIVE YOU

A confirmed role setup you can reuse to open future conversations, with Claude consistently applying your stated rules from the first message of each new session.

PRO TIP

Since this config doesn't persist automatically between separate conversations, save the whole setup prompt somewhere so you can paste it in fresh each time you start a new session in this role.

PROMPT #50

The Information Transformer

What it does: Transforms information from one format into a genuinely useful different format — not just reformatting, but restructuring for how the new format is actually used.

Best for: *Anyone who has information in one format and needs it usefully restructured into another*

COPY & PASTE INTO CLAUDE

Act as an information design specialist.

Source material: [PASTE THE INFORMATION – a transcript, raw data, a long email thread, notes]

Current format: [WHAT IT CURRENTLY IS]

Format I need it in: [WHAT YOU NEED – e.g. a checklist, a decision log, a structured brief, a table]

Why I need it in this format: [PURPOSE]

Transform it:

1. Don't just reformat – restructure the actual information around how the new format is meant to be used.
2. Preserve anything important from the source that the new format needs, and flag anything you had to leave out and why.
3. If the source material is missing something the new format typically requires, flag the gap instead of inventing content to fill it.
4. Deliver the final result in the exact requested format, ready to use as-is.

CUSTOMIZE THESE

Be specific about why you need the new format — a checklist meant for daily use and one meant for a one-time audit get structured differently, even from the same source material.

WHAT CLAUDE SHOULD GIVE YOU

A genuinely restructured (not just reformatted) output in your requested format, with anything cut clearly flagged and any real gaps in the source material called out instead of invented.

PRO TIP

If the output will be reused as a template, follow up by asking Claude to note which parts were specific to this instance versus reusable structure for next time.

BONUS SECTION

5 Claude Workflows You Should Save

These chain several prompts from this pack into a single end-to-end process. Run them in order, inside the same conversation, so Claude keeps the context from each step.

WORKFLOW 1

From Idea → Real Project

For turning a rough business/product idea into something scoped enough to build

THE CHAIN

- 1 Idea Weakness Finder (Prompt #39) — stress-test the idea honestly.
- 2 Target User Definer (Prompt #41) — narrow who it's actually for.
- 3 Customer Pain Point Extractor (Prompt #42) — confirm the real need, from real feedback if you have it.
- 4 MVP Scope Definer (Prompt #43) — cut it down to the smallest testable version.
- 5 Goal-to-Action-Plan Converter (Prompt #30) — turn the MVP into a concrete build plan.

KICKOFF PROMPT (paste this first)

```
I'm going to work through validating and scoping a business/product idea with you across several steps in this conversation. Here's the idea:
```

```
[DESCRIBE YOUR IDEA]
```

```
Start by acting as a skeptical but fair advisor and finding the real weaknesses in this idea – don't soften it to be encouraging. Once we've stress-tested it, I'll ask you to help narrow the target user, confirm the core pain point, scope an MVP, and build an action plan – all building on what we establish at each step.
```

WORKFLOW 2

From Beginner → Job Ready

For going from current skills to interview-ready for a specific target role

THE CHAIN

- 1 Skill Gap Analyzer (Prompt #19) — compare your skills against a real job description.
- 2 Study Plan Failure Diagnostician (Prompt #3) — if a learning plan for the gap stalls, diagnose why before pushing through it.
- 3 Portfolio Strategy Builder (Prompt #18) — decide what to build to prove the new skills.
- 4 Resume-to-Job-Description Matcher (Prompt #16) — make sure your resume shows what you now have.
- 5 Mock Technical Interviewer (Prompt #20) — practice under realistic pressure before the real thing.

KICKOFF PROMPT (paste this first)

```
I'm working toward a specific target role and want to go through the full readiness process with you across this conversation. Here's the job description I'm targeting:
```

```
[PASTE JOB DESCRIPTION]
```

```
My current skills/experience: [LIST YOUR SKILLS]
```

```
Start by analyzing the gap between my current skills and this job description, splitting gaps into critical and secondary. Once we know the gaps, I'll ask you to help me build a plan to close them, decide what portfolio work would prove I've closed them, check my resume against this job description, and finally run a mock interview.
```

WORKFLOW 3

From Bug → Understanding

For actually learning from a bug instead of just patching it and moving on

THE CHAIN

- 1 Error Root-Cause Diagnostician (Prompt #10) — systematically find the real cause.
- 2 Unfamiliar Code Explainer (Prompt #9) — if the buggy code isn't yours or isn't fully understood, get oriented first.
- 3 The fix, applied from step 1.
- 4 Test Suite Generator (Prompt #12 in Coding) — add a test that would have caught this bug.
- 5 A one-line “why this happened” note you keep for yourself — ask Claude to summarize the root cause and prevention in 2 sentences you can save.

KICKOFF PROMPT (paste this first)

```
I want to work through a bug with you in a way that helps me actually understand it, not just patch it. Here's the error and context:
```

```
Error: [PASTE ERROR/STACK TRACE]
```

```
Code: [PASTE RELEVANT CODE]
```

```
What I was doing: [GOAL/ACTION]
```

```
Start by working through this systematically — rank the most likely root causes with your reasoning, rather than guessing at one fix. Once we confirm the real cause and apply a fix, I'll ask you to help me write a test that would have caught this, and then summarize the root cause and prevention in two sentences I can save for later.
```

WORKFLOW 4

From Messy Notes → Mastery

For turning scattered notes into material you actually retain, not just organize

THE CHAIN

- 1 Notes-to-Learning-System Converter (Prompt #5) — structure the raw notes and surface gaps.
- 2 Concept Diagnostician (Prompt #1) — run on any concept flagged as a gap or that feels shaky.
- 3 Active Recall Quiz Builder (Prompt #2) — test yourself on the structured material.
- 4 Socratic Tutor (Prompt #6) — for the one or two concepts you still can't confidently explain after the quiz.

KICKOFF PROMPT (paste this first)

I have messy notes on a topic and want to turn them into something I actually retain, working through several steps with you in this conversation. Here are my notes:

[PASTE MESSY NOTES]

Start by turning this into a structured learning system — grouped by concept, with one-line summaries and any gaps in my notes flagged. Once we have that structure, I'll ask you to help me diagnose my understanding of any shaky concepts, quiz me with active recall questions, and use Socratic questioning on anything I still can't confidently explain.

WORKFLOW 5

From Long Document → Decision

For going from a long document you haven't fully read to a confident decision

THE CHAIN

- 1 Document Analysis Assistant (Prompt #46) — analyze it for your specific purpose.
- 2 Long-Document Actionable Extractor (Prompt #24) — pull out only what requires action or a decision.
- 3 Decision Pressure Test (Prompt #29) — weigh the actual options against your real priorities.
- 4 Assumption Challenger (Prompt #32) — stress-test your leaning before you commit to it.

KICKOFF PROMPT (paste this first)

I have a long document I need to make a decision based on, and want to work through it with you across several steps in this conversation. Here's the document and what I need to decide:

Document: [PASTE OR ATTACH DOCUMENT]

The decision I need to make: [DESCRIBE THE DECISION]

Start by analyzing this document specifically for what I need to decide — not a general summary. Once we've extracted what's actually actionable, I'll ask you to help me pressure-test the real options against my priorities, and finally challenge whatever I'm leaning toward before I commit to it.

KEEP BUILDING

More prompt packs at haas.dev

This pack is part of an ongoing library of practical resources for developers, students, and builders using AI to move faster — without losing the underlying skill. New prompt packs, curriculum PDFs, and guides are added regularly.

— **haas.dev**

Engineering mindset over tutorials.