Expanded AI Resource Library

Five subjects, three learning pages each

An expanded companion to the Free AI Resource Library. This edition turns the prompt collections, productivity guide, and cheat sheet into practical lessons, templates, and exercises.

Contents

1. ChatGPT Prompt Library - 3 pages
2. Claude Prompt Examples - 3 pages
3. Cursor Coding Prompts - 3 pages
4. AI Productivity Guide - 3 pages
5. Beginner AI Cheat Sheet - 3 pages

Use the templates as starting points. Replace bracketed text with your details, use only approved sources and tools, and review outputs before relying on them. Product capabilities and access can change by plan, platform, region, model, and workspace settings.

Prompt Library

Page 1 of 3 - Build a prompt that gets a useful first draft

Start with the job

A prompt is a working brief. Tell ChatGPT what result you need, who it is for, and how you will use it. “Write a short update for parents about a school event” supplies more direction than “write an announcement.” Use a direct verb such as explain, compare, extract, draft, plan, or check.

Add only context that changes the answer

Useful context may include audience knowledge, source documents, tone, geography, dates, approved facts, and constraints. Separate facts from preferences. If a number or date must stay unchanged, say so. If something is missing, ask the assistant to flag the gap instead of guessing.

Describe the shape of the result

Name the output form and the amount of detail: a checklist, table, one-page summary, lesson plan, or three options. State the reading level or tone if it matters. Avoid adding many fine-grained rules unless they protect an important requirement; overly long prompts can obscure the main goal.

Use a compact brief

For bigger tasks, use four pieces: goal, context, output, boundaries. Add a final check only when it helps, such as confirming every action has an owner. This structure is a guide, not a required syntax; short requests can stay short.

Copy-ready prompt

Goal: Create [result].
Context: [relevant background, audience, source material].
Output: [format, length, tone, level of detail].
Boundaries: Preserve [facts/requirements]; flag [unknowns]; avoid [unwanted content].
Before finishing, check [one important quality criterion].

Try it

Take a vague request such as “help with a presentation.” Rewrite it to include audience, purpose, number of slides, source material, and one boundary. Compare the responses and note which added detail made the biggest difference.

Page 2 of 3 - Use prompt patterns for common work

Writing and editing

For a draft, provide the audience, purpose, key facts, and tone. For editing, say what should change and what must remain. Ask for alternatives when choosing a voice. A safe instruction is to mark missing details with brackets rather than inventing them.

Research and current information

When a question depends on recent events or changing details, ask for web search if available. Request publication dates, direct sources, and a distinction between confirmed information and interpretation. Open the links yourself; a citation is a path to evidence, not proof that the statement is accurate.

Files and data

Attach only the file needed for the task. Tell ChatGPT what to extract or compare, which columns or sections matter, and how to report uncertainty. For spreadsheets, ask it to state how it interpreted fields, identify missing values, and show calculations or assumptions.

Learning and planning

For learning, request an explanation, example, and practice question; ask it to wait before showing answers if you want to test yourself. For project planning, request milestones, dependencies, assumptions, risks, and placeholder owners instead of fabricated names.

Examples to adapt

Research: “Find the latest official guidance on [topic]. Summarize what changed, include dates and links, and label anything you cannot verify.”
File: “From the attached report, extract [fields] into a table. Include page references and mark missing values as Not stated.”
Learning: “Teach me [topic] at beginner level with one example and three practice questions. Wait for my answers before giving solutions.”

Practice

Choose one real task from writing, research, files, data, learning, or planning. Write the prompt, identify which feature it depends on, and add a fallback request in case that feature is unavailable.

Prompt Library

Page 3 of 3 - Improve, verify, and use responses responsibly

Refine in small steps

Treat a first response as a draft. Give focused feedback: make the opening shorter, keep the evidence, move risks above background, or use a warmer tone. Point to the exact part to change. If the response misunderstood the task, restate the goal and supply the missing context rather than layering on unrelated corrections.

Check claims and sources

ChatGPT can produce fluent but inaccurate content. Verify important dates, calculations, names, policies, and quotations against trustworthy sources. For current information, inspect linked sources and check that they support the specific claim. If the answer affects health, law, money, safety, or an important decision, use qualified human review.

Protect information

Share the minimum material needed. Do not paste passwords, access tokens, or confidential material into a tool unless the workspace and approved policy permit it. Remove direct identifiers when they are unnecessary. Review the output for accidental disclosure before forwarding it.

Make the final check explicit

Before using a generated file or message, compare it with the original request: right audience, correct facts, complete sections, and appropriate tone. Ask for an uncertainty list or checklist when that helps. You remain responsible for deciding whether the result is ready to use.

Revision and review prompts

Revision: “Shorten this to 150 words. Keep the dates and decisions unchanged. Make the next steps easier to scan.”
Verification: “Check the draft against [requirements]. List unsupported claims, missing items, and conflicts. Do not silently fill gaps.”
Sources: “For each factual claim about [topic], show a reliable source and say what it supports.”

Mini exercise

Use an AI-generated answer from a low-risk topic. Highlight three factual claims, verify them using independent references, then revise the answer to distinguish supported facts from assumptions.

rompt Examples

Page 1 of 3 - Write clear instructions and give Claude the right context

Make the task concrete

Start with one primary action and a clear reason: summarize this brief for a new team member, compare these proposals for a budget decision, or draft a reply to a customer. State the audience and the decision or use the response should support.

Explain background an outsider would need

Give Claude the relevant definitions, project facts, constraints, and source material. Avoid relying on context that exists only in your head. If a document is authoritative, say so. If two sources conflict, ask Claude to report the conflict rather than select one without explanation.

Specify output and boundaries

Ask for the structure you can use: short memo, table, outline, annotated list, or draft. Include length and tone if important. Boundaries may include “use only these files,” “do not invent missing owners,” or “separate evidence from recommendations.” Keep the constraints proportionate.

Break complex work into stages

For a substantial task, request a first-pass plan or questions before the final deliverable. Then supply missing information, review the result, and ask for a focused revision. This makes it easier to spot a wrong assumption early.

Copy-ready prompt

I need [deliverable] for [audience/purpose]. Use [provided context or source]. Include [required sections]. Write in [tone/length]. Treat [source] as authoritative; identify conflicts or missing facts instead of guessing. First tell me any critical ambiguity, then create the draft.

Practice

Take a request you would give a colleague. Add the audience, what the colleague needs to know, what “done” looks like, and what should happen if evidence is missing.

Page 2 of 3 - Work with files, Projects, search, and Artifacts

Documents and uploads

When attaching a file, say what job it should perform: summarize, extract, compare, explain, or transform. Ask for page or section references where practical. Large or visually complex files may have format or processing limits, so verify that Claude is reading the portion you care about.

Projects for sustained work

Projects can organize chats and project knowledge such as supplied files and instructions, subject to plan and workspace availability. Provide durable information in the knowledge base or instructions rather than assuming every conversation automatically shares all details. Keep project instructions specific and update obsolete facts.

Web search for current facts

When web search is available, ask Claude to search for time-sensitive information and cite sources. Review each citation directly, compare multiple independent sources for consequential claims, and note the date of retrieval. Search availability and usage limits can vary.

Artifacts for standalone work

Artifacts can present a substantial document, diagram, or interactive creation separately from the conversation. Describe who will use it, what it should do, and how it should be tested. Iterate by naming the exact part to change and preserving elements that already work.

File and Artifact examples

File: “Compare the attached policies on [criteria]. Give a table with section references, conflicts, and unanswered questions. Do not infer policy.”
Project: “Use the approved project brief as the source of truth. Keep recommendations separate from recorded decisions.”
Artifact: “Create a simple [tool/visual] for [user]. Include [functions], accessible labels, and a short note on how to test it.”

Practice

Create a small project brief from approved, non-sensitive material. Ask Claude to produce a summary, then ask it to identify which statements are directly supported and which are inference.

rompt Examples

Page 3 of 3 - Examples for analysis, review, and iteration

Analyze without overclaiming

Ask Claude to show how it reached a summary or comparison: identify evidence, note the source location, and report limitations. For a decision, ask for options and trade-offs rather than a false sense of certainty. Separate “the source says” from “this may imply.”

Draft and revise

A useful writing loop is brief, draft, critique, and revision. First specify reader, purpose, tone, facts, and length. Then request a critique against your criteria. Finally ask for specific edits while preserving approved language or factual details.

Handle ambiguity explicitly

If a task depends on unknown requirements, ask Claude to list the questions that would materially change the answer. You can choose to answer them, have Claude state assumptions, or request a conservative draft with placeholders.

Use human judgment

Claude can help organize evidence and generate options, but the user should check consequential outputs. This includes verifying calculations, source links, code behavior, legal or medical claims, and external communications. A polished response is not the same as a validated response.

Prompts for iteration

Critique: “Review this draft against [criteria]. Identify the three biggest gaps and quote short phrases that need attention.”
Revision: “Apply only fixes 1 and 2. Keep the approved facts and commitments unchanged. Return the revised draft plus a short change note.”
Uncertainty: “List the assumptions you made, the evidence supporting each, and what I should confirm.”

Mini exercise

Provide a short public article or non-sensitive sample document. Ask for a summary, then challenge one statement and ask for its evidence. Record whether the follow-up improved traceability.

oding Prompts

Page 1 of 3 - Give the coding assistant intent and state

Intent describes the change

Say what should happen, who needs it, and how success will look. “Add password reset” is too broad without requirements. Specify visible behavior, inputs, expected result, and non-goals. For a small change, ask for a narrow edit; for a complex one, ask for a plan first.

State describes what exists now

Provide the relevant file, function, error, test failure, screenshot, or reproduction steps. In Cursor, focused references such as a file or symbol can direct the assistant to useful code. Too little context can lead to guesses; too much unrelated context makes key details harder to see.

Use Ask before Agent when exploring

When you are learning a codebase or deciding how to proceed, use a read-only question or planning workflow. Ask what files are involved, what patterns exist, and what risks matter before asking for edits. For implementation, make the requested scope and checks explicit.

Use rules for recurring conventions

Project rules can preserve durable guidance such as test commands, formatting, architecture choices, and areas that must not change. Keep them specific and current. A rule is useful shared context; it does not replace a clear task prompt or review.

Copy-ready codebase brief

Goal: [desired behavior]
Current state: [relevant files, logs, reproduction steps]
Constraints: [interfaces, style, files to avoid]
Acceptance criteria: [observable checks]
First: inspect and explain the plan. Then make the smallest suitable change, run relevant checks if available, and summarize the diff.

Practice

Open a small project. Ask Cursor to explain one feature without editing. Compare its file references with your own understanding, then write acceptance criteria for a small, safe change.

oding Prompts

Page 2 of 3 - Prompt patterns for coding tasks

Implementation

Describe one behavior at a time. Include inputs, outputs, edge cases, existing patterns, and expected tests. Ask the assistant to inspect first, implement a focused change, and report affected files. Avoid vague demands like “make it better” unless you ask for options first.

Debugging

Provide the exact error, steps to reproduce, expected behavior, actual behavior, and relevant logs. Ask for a root-cause explanation before a fix. A good debugging request discourages symptom-only changes and asks for a regression test when appropriate.

Tests and refactoring

State which behavior must be covered and which public behavior must stay the same. Ask for test changes separately when you want to inspect the plan. For refactors, preserve interfaces and require the existing tests or checks to run where practical.

Documentation and review

Ask documentation work to match existing project conventions and verify examples against code. For a review request, ask for concrete findings with locations and impact, not a generic approval. If no findings are found, ask the assistant to state the areas it reviewed.

Prompt examples

Debug: “Reproduce [error] using [steps]. Explain the root cause before editing. Make the smallest fix and add a regression test for [case].”
Test: “Add tests for [normal/edge/failure cases] using the existing framework. Do not alter production code unless the tests reveal a defect.”
Review: “Review this diff for concrete bugs, regressions, and missing tests. Rank findings by severity and cite file/line locations. Do not edit.”

Practice

Select a bug from a sample project. Write a prompt with reproduction steps and expected versus actual results. Ask for a diagnosis only, then decide whether the proposed cause fits the evidence.

Page 3 of 3 - Review changes, commands, and risk

Read the diff before accepting

A diff shows what changed. Review every touched file for scope, correctness, unintended behavior, and secrets. Check whether generated code follows project patterns and whether tests cover the behavior. Ask for an explanation when a change is unfamiliar.

Understand terminal actions

An assistant may propose commands to inspect files, run tests, install packages, or modify data. Read the command and consider its effects before approving it. Be especially careful with destructive operations, network actions, credentials, migrations, and commands that affect production.

Test what matters

Run the checks appropriate to the change: targeted tests, type checks, linting, builds, or manual steps. Read failures rather than assuming the assistant fixed them. Ask it to distinguish checks it actually ran from checks it recommends.

Recover and document

Use version control and available checkpoints as recovery tools, not as substitutes for review. Keep a concise record of purpose, files touched, tests, and known limitations. For a team, update durable project rules only after a convention is agreed.

Safe finish prompt

Before finishing, summarize the behavior changed, files touched, commands actually run, test results, and remaining risks. Do not claim a check passed unless it ran successfully. Flag any command or dependency change that needs my review.

Mini exercise

Ask Cursor for a small code edit in a non-production project. Inspect the diff, run a relevant check yourself, and compare the result with the acceptance criteria you wrote.

Page 1 of 3 - Find work where AI can help

Start from friction, not from a tool

List repetitive work that consumes time: summarizing meetings, drafting routine updates, searching documents, categorizing requests, or preparing first-pass reports. AI is most useful when the task is frequent and reviewable, and when people can describe a good result.

Map the work as it happens

Write down the trigger, input, steps, decisions, tools, output, and handoff. Include exceptions: missing information, conflicting records, urgent cases, and sensitive material. This prevents an “automation” from silently skipping the difficult cases.

Choose AI assistance versus rules

Generative AI is useful for flexible language and interpretation. Rules-based automation is better for predictable actions such as moving a file after a known event or calculating a fixed value. A combined workflow can use AI to classify, then a deterministic rule to route, with a human handling uncertain cases.

Set the boundary of responsibility

Decide what the system may draft, retrieve, or change, and what requires a person. Start with low-impact assistance. Require review for customer commitments, financial transactions, hiring decisions, public claims, or changes to important records.

Workflow canvas

Task: [recurring job]
Trigger/input: [what starts it and what information is needed]
Current steps: [people, tools, decisions]
Output: [what good completion looks like]
Exceptions: [when to stop and ask a person]

Practice

Pick one weekly task. Estimate its frequency and time, list the steps, and mark which steps require judgment or permission. Identify one small step AI could draft or summarize without taking action.

Page 2 of 3 - Design a dependable AI workflow

Standardize the request

Create a template with a clear goal, approved sources, audience, format, boundaries, and a final check. Add examples only when they show a meaningful pattern. Use explicit placeholders for missing owners, dates, or facts.

Prepare source material

Use accurate, current, permissioned information. Name the authoritative source and avoid mixing outdated and current documents without labels. If the task relies on external information, decide whether web search or a connected source is available and appropriate.

Make outputs easy to inspect

Use structured fields, short headings, citations, or a checklist so a reviewer can find errors. Ask the model to separate source facts from recommendations. For repeated processes, store outputs in a consistent format and track versions.

Design for failure

Plan what happens when the model is uncertain, a file is missing, a source conflicts, or a tool is unavailable. A good process can stop safely, ask for clarification, and route the case to a person. Do not silently substitute a guess.

Example: notes to action list

Input: approved meeting notes. AI drafts action, owner, due date, and source phrase. Missing owners or dates become “Not stated.” A meeting participant reviews the list, corrects it, then shares it through the approved channel. The AI does not send messages or assign work automatically.

Practice

Write a prompt for one mapped task. Test it with a normal example and an incomplete example. Check whether it reports missing data, preserves facts, and produces the same structure both times.

Page 3 of 3 - Pilot, measure, and improve responsibly

Run a small pilot

Choose one workflow, a small group, and a limited period. Save a baseline before introducing AI: completion time, error rate, rework, and user satisfaction. Define who can stop the pilot and how issues are reported.

Evaluate quality and effort together

Measure the whole job, including briefing and review time. Compare accuracy, completeness, tone, and consistency against a human baseline. A faster draft that needs extensive correction may not save time. Record the types of errors, not just a single average score.

Protect data and access

Use approved tools and minimum necessary access. Avoid sending sensitive information to an unapproved service. Clarify retention, sharing, and connector permissions with the organization. Treat a generated answer as data that may need the same review as other work products.

Govern ongoing use

Assign an owner, document the purpose and limits, keep a feedback route, and revisit the workflow when products or policies change. Human oversight should increase when consequences are harder to reverse or the system performs poorly.

Pilot scorecard

Baseline: [time, error/rework, satisfaction].
Trial: [same measures plus reviewer effort].
Quality checks: [facts, omissions, exceptions, privacy].
Decision: keep, revise, or stop.
Owner and review date: [name/role and date].

Mini exercise

Define a one-week pilot for a low-risk task. Specify the baseline, success measure, reviewer, data boundary, and stop condition. Do not scale until the results and failure cases are understood.

AI Cheat Sheet

Page 1 of 3 - Core terms and a simple mental model

Prompt and context

A prompt is the instruction or goal you send. Context is the relevant information available to the model, such as conversation messages, attached files, or connected sources. Better context can help, but irrelevant context can distract. Say what information should be used.

Model and tokens

A model generates outputs from patterns learned during training and the context it receives. Text is processed in smaller units often called tokens. The context window is a practical limit on how much input and conversation a model can use at one time; exact limits vary by product and model.

Hallucination and uncertainty

A hallucination is an answer that sounds plausible but is wrong or unsupported. Fluency is not evidence. Ask for sources or assumptions, inspect citations, and verify claims that matter. Models can also miss information, misunderstand a file, or produce different answers to similar prompts.

Tools, agents, projects, artifacts

A tool adds a capability, such as search or file analysis. An agent can use available tools across multiple steps toward a goal. A project organizes related work in some products. An artifact is a standalone generated output. Availability and behavior depend on the specific product and settings.

Quick glossary

Prompt = your request. Context = relevant material. Model = the AI system. Hallucination = plausible but possibly false output. Tool = extra capability. Agent = multi-step tool-using assistant. Project = organized work area. Artifact = reusable standalone output.

Check your understanding

Explain the difference between a prompt and context in your own words. Then give an example of a claim you would verify before relying on it.

AI Cheat Sheet

Page 2 of 3 - Write, refine, and choose the right task

Use a practical prompt recipe

State the job, the information the model should use, the audience, and the form of the answer. Add boundaries such as “use only these notes” or “flag unknowns.” A prompt can be one sentence for a simple question or a short brief for a larger task.

Choose suitable work

AI can help explain, brainstorm, summarize, draft, compare, extract, and analyze material when the right feature is available. It can assist with code and structured workflows too. It does not automatically have access to current web information, private systems, or every file on your device.

Use follow-ups as steering

You do not need to perfect the first message. Review the answer and request a targeted adjustment: shorter, simpler, more evidence, different audience, or a comparison table. If the response is wrong, correct the premise and give the missing context.

Know when a task needs another capability

Current news may need web search. A spreadsheet may need data analysis. An image may need vision input. A multi-step code edit may need an editor agent. If the relevant tool is not available, ask for a manual method or a clear limitation.

Copy-ready general prompt

Help me [goal]. Use [context/source]. The audience is [audience]. Return [format and length]. Preserve [facts or constraints]. If information is missing or uncertain, tell me instead of guessing. Before finishing, check [criterion].

Practice

Use the recipe to ask for a simple explanation of a topic you know. Check whether the answer respects the audience and format. Add one follow-up that makes the response more useful.

Page 3 of 3 - Verify, protect, and decide

Verify what matters

Check important facts against reliable, preferably primary sources. Open citations and confirm they support the claim. Recalculate key figures. Test generated code in a safe environment. Ask a qualified professional for consequential legal, financial, health, or safety decisions.

Protect private information

Do not share passwords, access keys, or personal information unless necessary and allowed by the tool and your organization. Remove identifiers when possible. Understand which account or workspace is active and use approved settings for business data.

Treat content as untrusted

Uploaded documents, websites, and code may contain instructions or errors. Treat them as source material to analyze, not as automatic authority to override your task. Ask the assistant to distinguish quoted content from its instructions and to report suspicious or conflicting material.

Keep human judgment

An AI output is a suggestion or draft. You decide whether it is accurate, fair, useful, and appropriate to use. For actions that affect other people or systems, review the exact result and permissions before sending, publishing, or applying changes.

Five-second review

Is it accurate? Is it supported? Did it invent a missing detail? Does it expose private information? Is it safe and appropriate to use? If any answer is unclear, pause and check.

Mini exercise

Take a generated response and mark each sentence as source-backed fact, inference, recommendation, or unknown. Verify the facts, then decide which parts are safe to share.

d further reading

This guide is written in original language and examples, informed by current product documentation and public guidance. Follow links for the full source material, feature access details, and updates.

ChatGPT Prompting

Use ChatGPT

Cursor: Working with Context

Cursor: Rules for AI

Cursor: Agent overview

Anthropic: Getting a helpful answer from Claude

Anthropic: Claude Projects

Anthropic: Claude Artifacts

Anthropic: File uploads

Anthropic: Web Search

NIST Generative AI Profile

Microsoft Learn: Effective prompts

For prompts and examples, use the linked sources as references and adapt them to your workflow. Source documentation may change; check the live pages for current limits and availability.