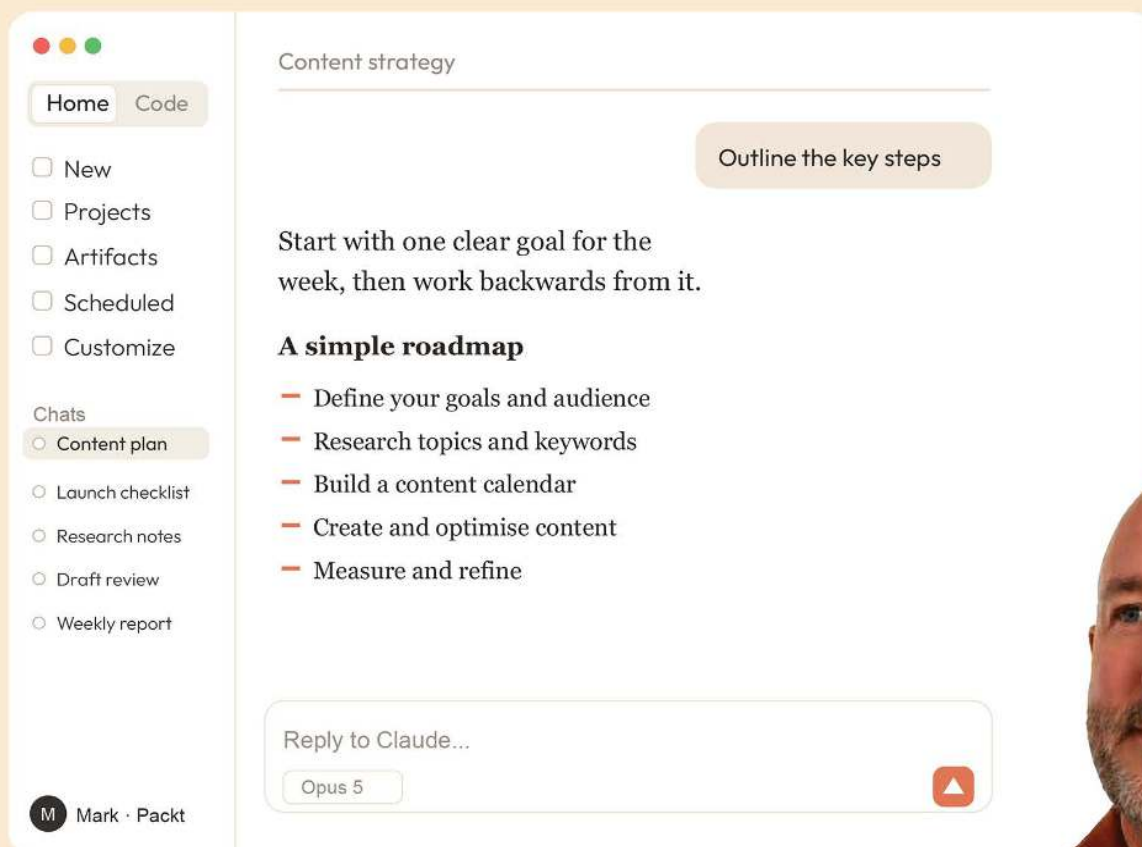


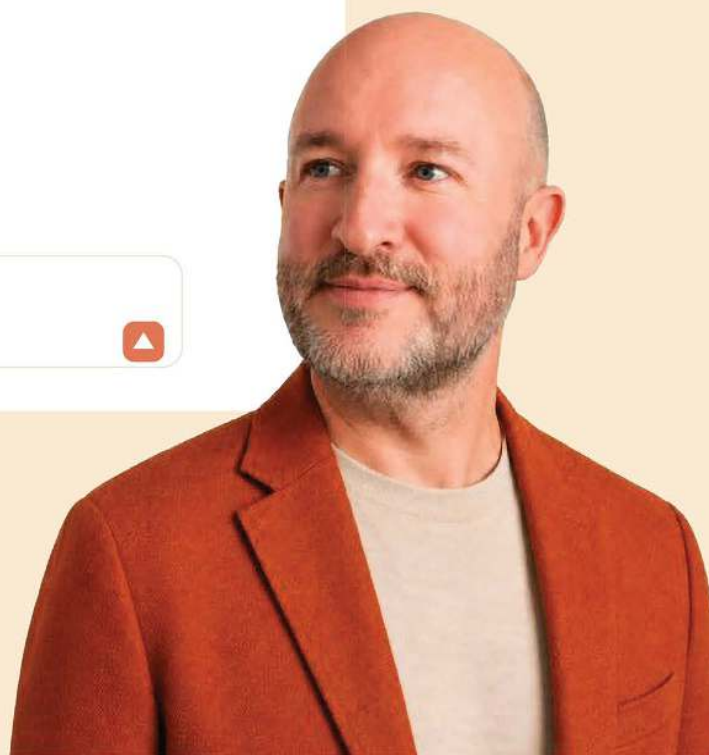
Claude

Visual Bible

A step-by-step, screenshot-by-screenshot guide to writing, researching, analyzing, and automating with Claude



Mark J. Price



Claude Visual Bible

A step-by-step, screenshot-by-screenshot guide to writing, researching, analyzing, and automating with Claude

Mark J. Price



Claude Visual Bible

Copyright © 2026 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Portfolio Director: Ashwin Nair

Relationship Lead: Nitin Nainani

Project Manager: Ruvika Rao

Content Engineer: Hayden Edwards

Technical Editor: Arjun Varma

Copy Editor: Hayden Edwards

Production Designer: Ajay Patule

Growth Lead: Sayantani Saha

First published: September 2026

Production reference: 1160926

Published by Packt Publishing Ltd.

Grosvenor House

11 St Paul's Square

Birmingham

B3 1RB, UK.

ISBN 978-1-80865-839-6

www.packtpub.com

Contributors

About the author

Mark J. Price helps beginners and everyday knowledge workers get real, practical value out of AI tools like ChatGPT and Claude. Based in Brighton, England, he brings two decades of experience as a technical author, including ten editions of a bestselling C# and .NET series for Packt, to making AI approachable for readers with no technical background. A former maths teacher and Microsoft Certified Trainer, he comes from a family of teachers and now chats with readers in Packt's Discord.



Thank you to all my readers. Your support means I get to write these books and celebrate your successes. Special thanks to the readers who give me actionable feedback via my GitHub repository and email, and interact with me and the book communities on Discord. You help make my books even better with every edition.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book or have any general feedback, please email us at customercare@packt.com and mention the book's title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you reported this to us. Please visit <http://www.packt.com/submit-errata>, click **Submit Errata**, and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit <http://authors.packt.com/>.

Free benefits with your book

This book comes with free benefits to support your learning. Activate them now for instant access (see the “*How to Unlock*” section for instructions).

Here’s a quick overview of what you can instantly unlock with your purchase:



DRM-Free PDF Version

Download DRM-free PDF and ePub copies of this book.



7-Day Packt Library Access

Get 7-day unlimited access to 8,000+ books and videos. No credit card required.

Available for first-time Packt+ trial users only.



Next-Gen Reader Access

Read this book on Packt Reader with progress sync, dark mode and note-taking.

How to Unlock

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.



UNLOCK NOW

Note: Keep your invoice handy. Purchases made directly from Packt don't require one

Table of Contents

Introduction	1
Why you picked up this volume	1
How this volume is structured	4
How to read the four books	5
Get the sample files and bonus content	5
A note about the figures in this book	6
How this book handles a fast-changing Claude	7
Why not just ask Claude to explain itself?	8
Key terms to help you follow along	9
What prerequisites do you need?	10
Book 1: Ask Claude	11
Chapter 1.1 Understand What Claude Can Do	12
Chapter 1.2 Create Your Account and Choose a Plan	18
Chapter 1.3 Use Claude on the Web, Phone, and Desktop	23
Chapter 1.4 Have Your First Useful Conversation by Typing, Talking, or Showing	30
Chapter 1.5 Write, Refine, and Improve Prompts Through Conversation	35
Chapter 1.6 Learn Something New or Tutor a Child	41
Chapter 1.7 Plan Everyday Life and Compare Important Choices	47
Chapter 1.8 Search and Verify Current Information	50
Chapter 1.9 Conduct Research with Cited Sources	53
Chapter 1.10 Add Documents and Images to a Chat	56
Chapter 1.11 Organize with Instructions, Memory, and Projects	59
Chapter 1.12 Project – Build Your Personal AI Workbench	64
Book 2: Create Real Work Results	69
Chapter 2.1 Use Claude Safely and Responsibly at Work	70
Chapter 2.2 Choose the Right Mode for Each Task	73
Chapter 2.3 Draft Professional Documents	79
Chapter 2.4 Summarize and Extract Information	86
Chapter 2.5 Analyze Spreadsheets and Data	91

Chapter 2.6 Create Visual and Interactive Content	98
Chapter 2.7 Create Presentations and Visual Reports	104
Chapter 2.8 Add Plugins, Connectors, and Skills	110
Chapter 2.9 Project – Produce a Useful Weekly Briefing	118

Book 3: Automate Recurring Tasks **125**

Chapter 3.1 Choose the Right Automation Building Block	126
Chapter 3.2 Schedule Recurring Work	128
Chapter 3.3 Turn a Procedure Into a Reusable Skill	135
Chapter 3.4 Build a Library of Skills for Real Work	149
Chapter 3.5 Connect Trusted Information and Tools with MCP	155
Chapter 3.6 Control Software with Computer Use	162
Chapter 3.7 Automate Personal Organization and Build Workflows	172
Chapter 3.8 Project – Build and Measure an End-to-End Workflow	177

Book 4: Advance Your Career and Income **181**

Chapter 4.1 Choose What Better Work Means to You	182
Chapter 4.2 Audit Your Role and Find High-Value Work	184
Chapter 4.3 Measure and Document Your Improvement	186
Chapter 4.4 Learn the Skills Your Next Role Requires	189
Chapter 4.5 Combine AI Fluency with Human Strengths	191
Chapter 4.6 Propose and Lead an AI Improvement at Work	193
Chapter 4.7 Build a Case for Promotion or Higher Pay	195
Chapter 4.8 Prepare for a Better Job or Career Change	198
Chapter 4.9 Build a Portfolio and Measurable Case Study	200
Chapter 4.10 Turn a Problem into a Service or Product	203
Chapter 4.11 Project – Complete a 30-day Career Value Project	205

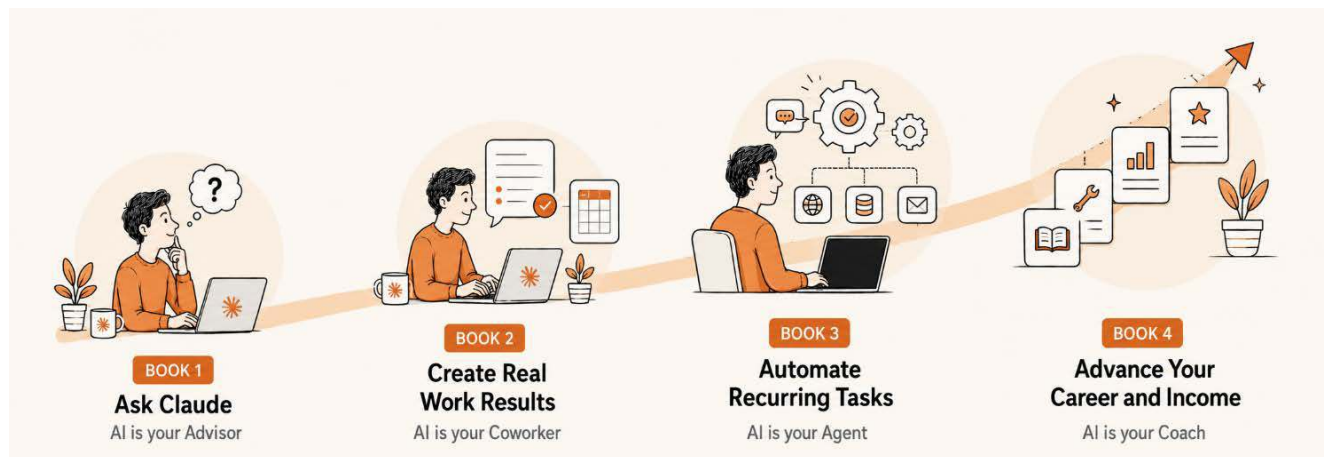
Epilogue **209**

Introduction

Welcome to *Claude Visual Bible*!

In this introduction, you will learn:

- Why you picked up this volume
- How this volume is structured
- How to read the four books
- A note about the figures
- How this book handles a fast-changing Claude
- Why not just ask Claude to explain itself?
- Key terms to help you follow along
- What prerequisites do you need?



Why you picked up this volume

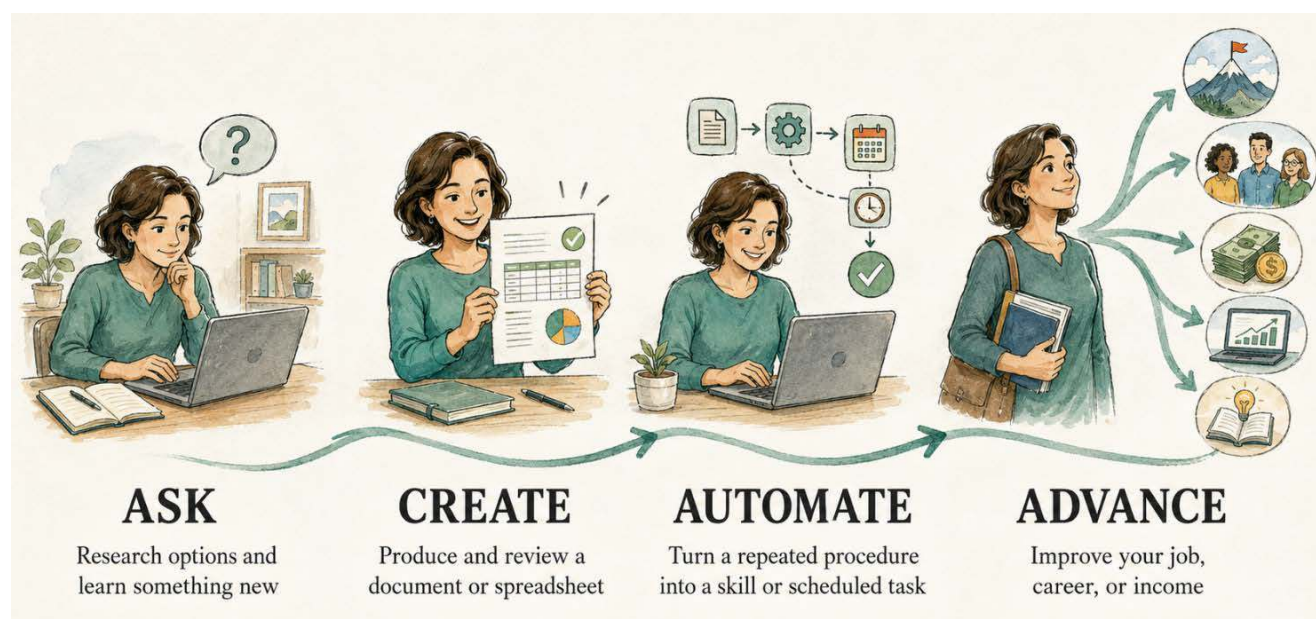
At this point in your journey with AI, you may have sensed that Claude could help you work faster and think more clearly. Maybe you have asked it a few questions, got it to rewrite an email, or asked for an idea when you got stuck. But you probably still treat it as a chatbot: something you type into, not something you create, do research, or automate aspects of your life with.

Closing the gap between casual, occasional Claude use and using it to genuinely improve your work and personal life, is why I wrote this volume. You do not need a technical background to close the gap. You need a clear path from your first conversation to confident, professional use, with checkpoints along the way that reinforce what you learn.

Most guides to Claude choose one of two extremes. Some try to cover several large subjects, including coding, in one slim volume, so each subject gets miserly treatment. Others promise easy income from a handful of prompts, without teaching you how to check whether the result is any good.

Neither approach matches how real people truly build value with AI. The real path looks like this: ask for useful guidance, produce a checked piece of work, turn a proven procedure into something repeatable, then use that evidence to improve a career or generate a new income.

Claude Visual Bible is different because it follows that better path, one step at a time:



Introduction Figure 1: Four steps to advance your work and home life with AI

This guide also gives you something other books don't: premade files to practice on. Every chapter includes sample files matched to that chapter's task, like a Word document to summarize, or a spreadsheet to analyze, so you can start immediately instead of hunting for your own test material or skipping the exercise altogether.

Alongside those files, tasks come with one or more prompts you can copy straight into Claude. You're not left to guess how to phrase a request or peer at tiny text in a poorly cropped screenshot. You get a working prompt as a starting point, then a clear explanation of which parts to tweak for your own documents, your own job, and your own goals.

Claude Visual Bible also takes a wider view of what building value with Claude means. The obvious move is selling a product or service, but that route has genuine risk and a low success rate. *Book 4* covers that path, but treats it as one option among several, alongside using Claude to work faster in your current job, build the case for a promotion, or prepare for and land a new career. For most readers, improving what they already do is a far more practical, more likely route to success than starting a business from nothing.

That practical scope extends past work, too. In *Book 1* you'll find tasks aimed at your personal life, such as planning, organizing, and budgeting at home, or helping your kids with their homework and teaching them how to use AI responsibly.

Who this book is for

This book is for you if you have the curiosity and the everyday computer skills most adults already do, such as using a web browser, installing an app, and working with documents, spreadsheets, and images.

You will get the most from this book if you recognize yourself in one or more of these situations:

- You want to close the gap between casual Claude use and confident, professional use

- You manage a team, a small business, or your own workload, and want to work faster without sacrificing quality
- You want to build a habit of checking AI output instead of trusting it automatically
- You want to use Claude for personal tasks too, such as planning, budgeting, or helping a child complete school assignments ethically
- You are looking for a path from your first conversation with Claude to a completed piece of work, a repeatable process, or a stronger case for a promotion or new role

A flesh-and-blood author writing for a respected publisher

I am a living, breathing human based in Brighton, on England's south coast. Close enough to London for exciting cultural weekends, breezy enough to survive the increasingly sweltering British summers.

I've written more than 20 programming titles for my publisher Packt since 2016, including ten editions of my best-selling C# and .NET book. I update it every year in November and I plan to do frequent updates for *Claude Visual Bible* too since Claude is improving so rapidly.

Teaching is in my blood. My mother was a maths teacher, my sister a primary school teacher, my grandfather taught at the Royal Military Academy Sandhurst, and teaching and military careers run through my uncles and cousins too. I taught GCSE and A-Level maths in two London secondary schools, and I hold a Computer Science BSc (Hons) from the University of Bristol.

That mix, maths teacher and software developer, turns out to be exactly what a beginner's book about AI needs. Teaching maths to teenagers taught me how to break an intimidating idea into the smallest step that still makes sense, and how to spot the exact place a student gets stuck before they even ask. Software development taught me precise problem solving: to test whether something actually works rather than assume it does, and to write instructions a computer, or a nervous beginner, can follow without guessing. Put those together and you get a guide that walks you from confusion to confidence one checked step at a time, the way a good maths teacher walks you from "I don't get it" to "oh, that's actually simple."



Introduction Figure 2: Me and my background

There's another reason my software developer background matters here, beyond teaching precision: programmers are the canary in the coal mine for what AI does to a profession. I've spent the last few years watching AI move from autocomplete to something that can draft, test, and ship real code. That's a transition I lived through in my own daily work, which is exactly why I can show you, concretely, what to watch for as the same pattern reaches and transforms your profession.

Writing books for a decade took my face-to-face classroom teaching instinct further: it's how I now reach far more people, in far more places, than any classroom could, and I love hearing from readers all over the world. It's my way of making a dent in the universe, as Steve Jobs mused he wanted to do.

My involvement didn't end when I delivered the final drafts to my publisher. I am active in Packt's Discord channels, where you can chat with me and other readers about my books and share tips and tricks on how to get the most out of AI.

You can learn more about me and confirm my credentials at the following links:

- GitHub repositories: <https://github.com/markjprice/>
- LinkedIn: <https://www.linkedin.com/in/markjamesprice/>
- Amazon: <https://www.amazon.com/stores/Mark-J.-Price/author/B071DW3QGN>
- Packt: <https://subscription.packtpub.com/search?query=mark+j.+price>
- Discord channel: <https://discordapp.com/channels/963343534323269662/963344584287604736>

How this volume is structured

This single volume contains four books. Each one covers a stage of that path:

Book title	AI role	What changes for you
<i>Book 1 – Ask Claude</i>	Advisor	You move from asking occasional questions to holding a useful conversation, doing real research, learning new skills, and organizing your work using Claude features like projects.
<i>Book 2 – Create Real Work Results</i>	Coworker	You produce documents, spreadsheets, presentations, analysis, visual and interactive content, and other media. You learn how to prepare briefs for images that need a dedicated image-generation tool, and you learn to supervise longer, multi-step assignments.
<i>Book 3 – Automate Recurring Tasks</i>	Agent	You turn a procedure you repeat often into a skill or a scheduled task that Claude can run for you.
<i>Book 4 – Advance Your Career and Income</i>	Coach	You use what you learned in the other books to earn a promotion, make a career change, or provide a service to generate additional income.

Each of the four books in this volume contains a series of concise chapters and ends with a practical hands-on project, so you finish every book holding a completed piece of work, not just a lot of theory swimming around in your head.

You pay Anthropic only when you're ready to use what their paid subscription plan gives you. You can complete nearly all of *Book 1* on the **Free** plan. As your needs grow into real work and repeatable automation in *Book 2* and *Book 3*, you can upgrade your plan to **Pro** or higher which covers everything you'll need: Claude Cowork, scheduled tasks, connectors, skills, and plugins. Only features like Claude Tag, admin-built plugins, and org-wide connector authorization sit beyond an individual plan, reserved for Team- and Enterprise-managed workspaces most readers won't need.



Watch out: Several of the listed features are still in beta at the time of writing and could get removed or get renamed after publication.

How to read the four books

If you are new to Claude, then read the four books in order. Each one builds on the skills from the one before it. You cannot automate a procedure in *Book 3* until you can produce reliable work in *Book 2*, and you cannot produce reliable work until you can hold a useful conversation, which is what *Book 1* teaches.

If you already use Claude regularly, start closer to where your own skills stop. Skim the chapter headings at the start of *Book 1* and *Book 2*. If the tasks look familiar, move ahead to *Book 3*.

Inside each chapter, look for several types of recurring boxes:

- **Watch out** boxes warn you about a specific mistake or risk
- **Good practice** boxes give you a habit that improves your results
- **What to ask next** boxes give you one optional follow-up prompt you can send right after finishing the chapter's task
- **Real-world example** boxes give you a need solved by real people using AI

Two more boxes point you outside the book:

- **Learn more online** boxes link to an official resource, such as documentation from the AI provider. Interfaces and features change often. When a detail is likely to change after you buy this book, I give you a link to a verified source instead of guessing.
- **Sample files** boxes link to a file or folder in this book's online repository. Download these files when a chapter's activity needs them, if you don't want to use your own. You can open them, upload them into Claude, or copy from them, depending on what the chapter asks you to do.

You don't need to use every box. Read the ones that help with what you're doing right now.

Each chapter ends with a **Try it now** section that gives you one task you can finish in five to fifteen minutes. It also gives you a few questions to check what Claude produced, so reviewing its work becomes a habit rather than an afterthought.

Every **Try it now** task is written to work for as many readers as possible, so it stays intentionally general rather than a task built around one specific job. That breadth is what lets a teacher, an accountant, and a shop manager all work through the same chapter. If you'd like a task closer to your own role, I keep a growing online list that matches common professions to the chapter where a suggested, tailored prompt fits, so you can try something that maps directly onto your day-to-day work instead of adapting the general version yourself. Find it at <https://github.com/markjprice/claude-vb/blob/main/docs/bonus/professions/README.md>. I add new professions and prompts to that list over time, so it's worth checking back even after you've finished the book.

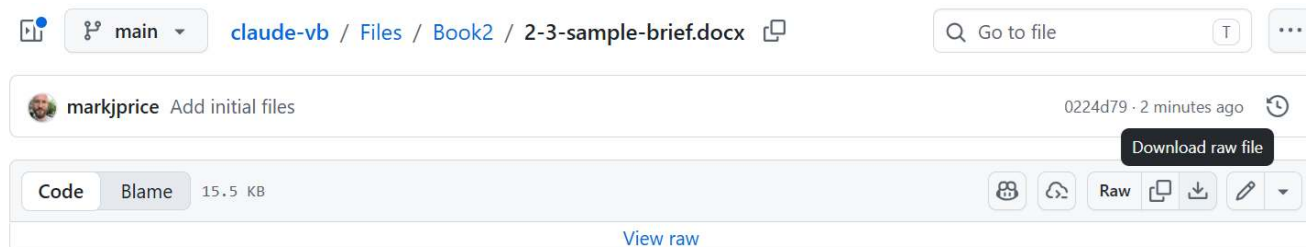
Get the sample files and bonus content

Other Claude guides expect the reader to supply their own files to work with while learning skills. *Claude Visual Bible* provides sample files, so you don't have to scramble to think of or create your own. These files are stored in a repository in my GitHub account so I can keep them updated.

Don't worry! Although GitHub is designed for software developers, anyone can use it. For example, you can go to the following link to see sample files you could use in *Book 2*:

<https://github.com/markjprice/claude-vb/tree/main/Files/Book2>

Just click a file like `2-3-sample-brief.docx` and then click the **Download raw file** button:



Introduction Figure 3: Download a sample file from this volume's GitHub repository

You can also download all the folders and files in one go using the following link:

<https://github.com/markjprice/claude-vb/archive/refs/heads/main.zip>

Once the ZIP file is downloaded, extract its contents to your local computer folder.

I have also written three accompanying bonus books:

- *Beyond Your First AI*: This bonus book explains when a different AI service is the better tool for a specific job, or a backup when your primary AI is unavailable. It covers OpenAI ChatGPT, Google Gemini, Mistral Vibe, and some Chinese providers like DeepSeek and Qwen.
- *Profession-Specific Prompts*: This bonus book adds a layer that the general prompting skills taught in the print books do not cover: the specific duties, risks, and rules that come with 20 professions.
- *Local AI Models*: The print books taught you to work with Claude running in Anthropic's cloud. This bonus book covers running an AI model entirely on your own laptop or desktop, with nothing leaving your device. Since Anthropic does not release any of its own models, this uses free and open-weight models like Google Gemma 4, Alibaba Qwen3.8, Meta Muse Glimmer, and NVIDIA Nemotron 3.5 Lightning.

You can find links to these in the *Epilogue*.

A note about the figures in this book

The word *Visual* in *Claude Visual Bible* is a promise. Every screenshot and diagram does one specific job: it helps you decide what to do, shows you what Claude is doing, walks you through an action, or helps you judge whether a result is good enough to use.

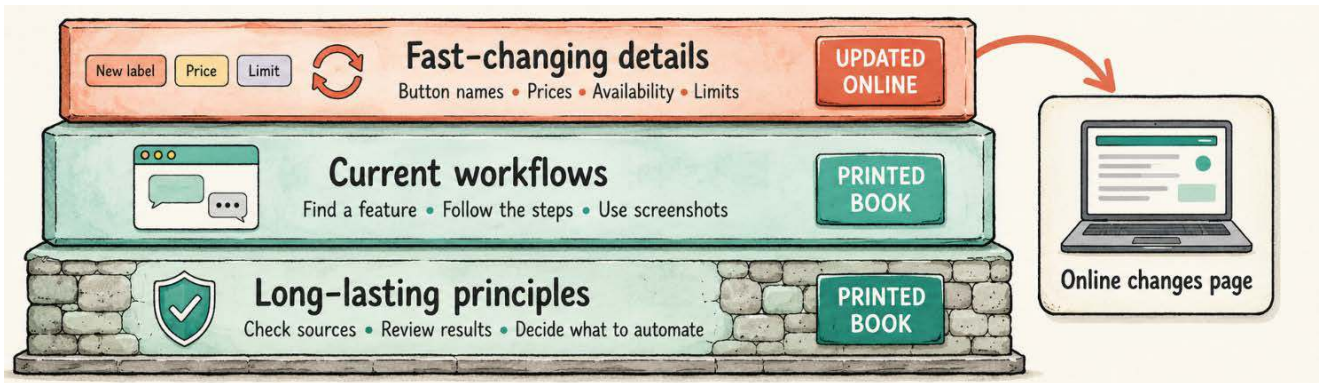
- Screenshots are cropped to the part of the screen that matters and enlarged so the interface text stays readable on a printed page. Where a screenshot shows several controls, I number them and explain each one in order.
- Diagrams are reserved for ideas that are hard to explain in a sentence, such as how a project holds context, or how a process moves from a trigger to a finished result.

I never place essential information only inside an image. When a figure matters to a task, the surrounding text explains it too, so the instructions still work even if a picture is hard to see.

One of my favorite benefits offered by my publisher is that for every purchase, Packt provides free PDF and online web versions. You can unlock your free benefits by turning to the *Free benefits with your book* section in the book's front matter. This will allow you to zoom into full-color screenshots if that helps.

How this book handles a fast-changing Claude

Claude's menus, buttons, and features change often. Books built only around today's exact user interface would be out of date within months.



Introduction Figure 4: A solid foundation, current screenshots, with fast-changing details updated in an online changes page in the book's GitHub repository

So, the books teach decisions and tasks that stay useful after the interface moves: how to judge whether a source is reliable, how to decide whether a task belongs in a quick chat or a longer piece of work, how to tell when a procedure is stable enough to automate. These long-lasting principles will serve you well while Claude evolves rapidly over the coming months and years.

Screenshots illustrate these ideas using the interface as it exists today, and each set of instructions combines a stable landmark, such as a menu's purpose, with the exact label currently shown on screen. If a label changes, the landmark still points you to the right place.

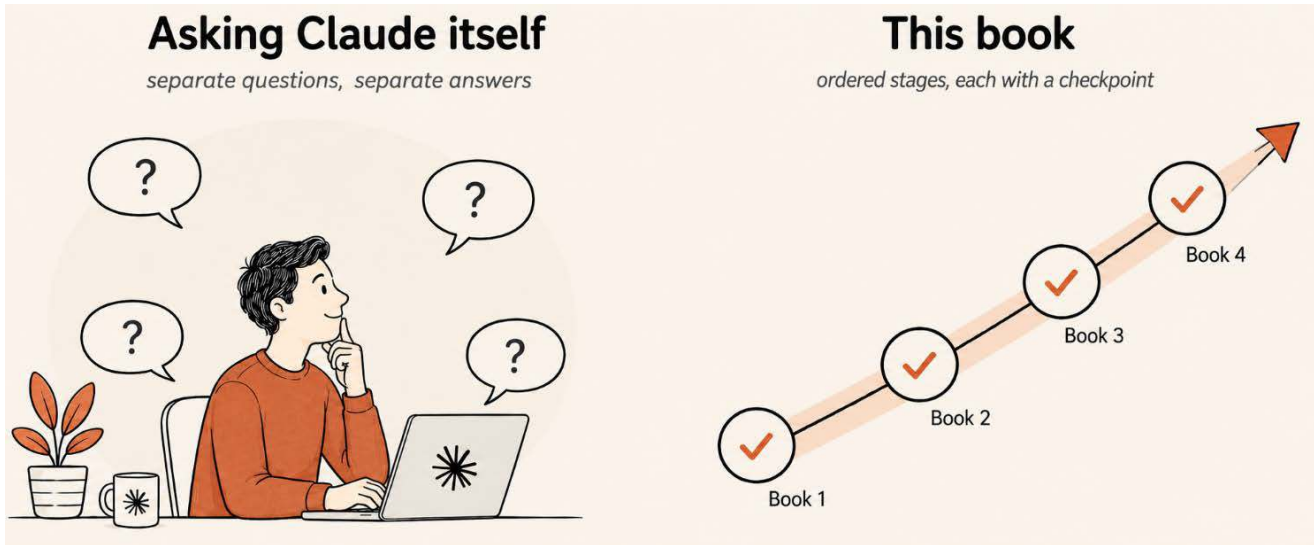
Where a detail is likely to move, such as plan prices or a settings screen, I have placed it on a companion page you can check online instead of printing it here. I will keep that page updated after the book is printed, so the printed pages can stay focused on what to decide and why, not on details a future software update could make wrong.

Visit <https://github.com/markjprice/claude-vb/blob/main/changes.md> for the current list of changes. I organize it by chapter number, so you can jump straight to the section that affects the page you are reading. This page shows:

- **Interface changes:** A new menu name, a moved button, or a renamed setting, matched to the chapter and step where the old version appears in this book.
- **Plan and pricing changes:** Current plan names, prices, and usage limits, since these move more often than the underlying features.
- **Feature availability:** Notes on when a feature described as a preview becomes generally available, or when a feature is removed.
- **Region and account differences:** Cases where a feature works differently, or not at all, depending on your country, device, or account type.
- **Corrections:** Any error identified in the book after printing.

Why not just ask Claude to explain itself?

You can ask Claude to explain its own features, and you should keep doing that after you finish this book. It is a fair question to ask why you need a book at all. Here is what this book adds:



Introduction Figure 5: Comparing asking Claude itself and reading this volume

Ask Claude four separate questions and you get four separate answers, each shaped by how you happened to phrase the question that day. This book orders material into four stages, each building on the skills from the one before it, and each ending with a project that gives you something finished to show for it. This book is a curated experience crafted by an education expert.

Claude can describe a feature confidently and still be wrong about whether that feature is available on your plan, in your region, or on your device. This book states those limits where they matter and points you to a page I keep updated after printing.

Claude can check or critique its own work when you ask, but that is not independent verification. The books therefore build a human check into every chapter, with a short Check the result section that asks you to review accuracy, sourcing, and safety before you rely on anything Claude produced. The books have been checked against primary sources and reviewed before publication. A chat response has not. I verify product details against official documentation, mark anything uncertain, and correct errors on the companion update page.



Good practice: Use Claude with web search for quick, current answers. Use this book for the parts that need structure, verification, and a plan you can follow to the end.

Key terms to help you follow along

This glossary covers the terms used most often throughout the books:

Term	Definition
Artifact	Substantial, standalone content that Claude creates in a dedicated panel separate from the conversation, such as a document, code, diagram, visualization, webpage, or interactive tool. You can review and revise an Artifact without losing your place in the chat. Claude also has a separate file-creation capability for producing downloadable Word, Excel, PowerPoint, and PDF files.
Chat	The Claude mode built for questions, discussion, and shorter, focused assistance.
Code	The Claude mode built for software development: writing code, running tests, and working with Git repositories. This book does not teach Claude Code.
Connector	A connected service that gives Claude access to information or actions outside the conversation, such as a calendar or a file storage service.
Context	Information you give Claude, through your message, an uploaded file, or an earlier part of the conversation, that shapes its response.
Cowork	The Claude mode built for longer assignments: research, connected files and apps, and finished documents, spreadsheets, presentations, or reports.
Dictate	A way of entering text by speaking instead of typing. Claude transcribes what you say into the message box; you are still writing text, just with your voice instead of your hands. Not the same as Voice.
Incognito Chat	A conversation you turn on that Claude does not save to your history and does not use to create memories.
Instructions for Claude	Personal preferences and background, such as your role or preferred tone, that you save once in Settings so Claude applies them automatically to every new conversation without you repeating them. Projects can also have their own instructions that apply only to that project.
Memory	Information Claude can store between conversations. You can review, correct, or delete it at any time.
MCP (Model Context Protocol)	A standard way to connect Claude to external tools and information, so different services can work with Claude without a custom-built connection for each one.
Plugin	An installable bundle that can include skills and connectors for a specialized task.
Project	A workspace that keeps related chats, files, knowledge, and instructions together around one ongoing piece of work.
Prompt	The message you send Claude: what you want it to do, and what a good answer should include.

Research	A Claude capability that plans a research task, gathers sources, and produces a longer, cited report instead of a single quick answer.
Scheduled task	A set of instructions that Claude runs automatically at a time or interval you choose, without you starting the conversation yourself.
Skill	A reusable set of instructions, examples, and resources that Claude can run to complete a specific, repeated task.
Voice	A mode where you speak to Claude and it replies in an ongoing spoken conversation, rather than only transcribing your words into text. Not the same as Dictate.
Web search	Searches the internet for current information and includes what it finds, with citations, directly in its response. Use it for a single question with a quick, current answer. Use Research (see above) when the question needs a longer, multi-source report instead of one search.

What prerequisites do you need?

You need very little to start *Book 1*:

- Claude is available across Windows, macOS, Chromebook, Linux, iOS, and Android, either through a web browser or a dedicated app. The exact features can differ by device, but you can use whichever device you already have to begin *Book 1*.
- A free Claude account. You can complete nearly all of *Book 1* on the free plan, although some features, such as Research, require a paid plan.
- A Claude Pro subscription, or higher, for the paid features like Research. By *Book 2*, I recommend upgrading to Pro or higher so you can follow the activities that use Cowork and other paid capabilities.
- Familiarity with common documents, spreadsheets, and presentations. You don't need specific software. Sample files for every chapter are ready to download from this volume's GitHub repository, so your own files are optional, not required.

A few more things will help before you turn the page:

- *Claude Visual Bible* covers Claude's Chat and Cowork experience on the web, phone, and desktop. It does not teach software development. If your goal is to write code, or build websites or mobile apps, look for the companion title, *Claude Code Visual Bible*.
- Treat every Claude response in this book as a draft to review, not a finished answer to trust automatically. Each chapter shows you what to check. A wrong answer stated with confidence can still cost you time, money, or credibility, so this habit is worth building early and often.
- Keep a notebook, document, or note app open as you read. Several chapters ask you to try a short task, and you will want somewhere to keep what you produce for the project at the end of each book.

With that, you are ready to open *Book 1* and have your first useful conversation with Claude.

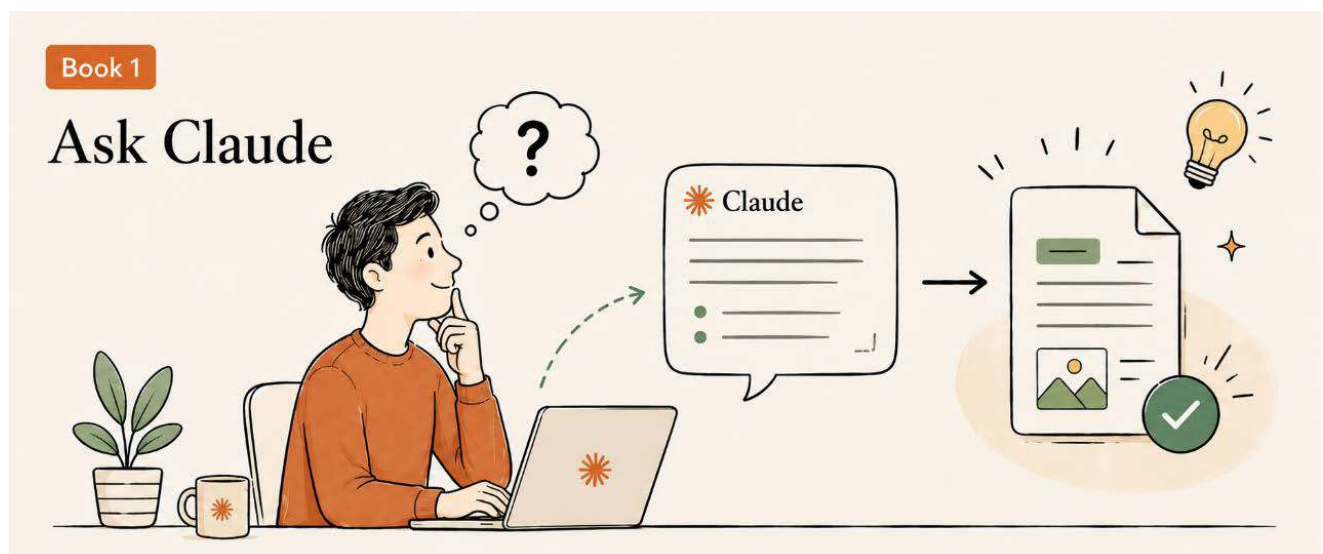
Book 1

Ask Claude

Book 1 is about moving from asking occasional questions to holding a useful conversation with AI as an advisor and mentor, doing real research, learning new skills, and organizing your work in projects.

It has twelve chapters, including an end-of-book project:

- 1.1 Understand What Claude Can Do 12
- 1.2 Create Your Account and Choose a Plan 18
- 1.3 Use Claude on the Web, Phone, and Desktop 23
- 1.4 Have Your First Useful Conversation by Typing, Talking, or Showing 30
- 1.5 Write, Refine, and Improve Prompts Through Conversation 35
- 1.6 Learn Something New or Tutor a Child 41
- 1.7 Plan Everyday Life and Compare Important Choices 47
- 1.8 Search and Verify Current Information 50
- 1.9 Conduct Research with Cited Sources 53
- 1.10 Add Documents and Images to a Chat 56
- 1.11 Organize with Instructions, Memory, and Projects 59
- 1.12 Project – Build Your Personal AI Workbench 64





Download the PDF version of this book

Your purchase includes a DRM-free PDF copy of this book, the code bundle, and a range of exclusive benefits. To unlock everything, follow the *Free benefits with your book* section in the front matter.

Chapter 1.1 Understand What Claude Can Do

You can ask Claude a question and receive an answer in seconds. That conversation is useful, but it is only one way to work with the product.

This chapter will show you how a useful conversation can grow into research, a finished deliverable, and a repeatable work process.

From chatbot to work platform

The back-and-forth exchange of chat messages that you are already familiar with is still at the center of Claude, hence the name of **Chat** mode. But Claude can do much more. Depending on your plan, you can speak, show it a photo, upload a document like a spreadsheet, search the web, or connect information from another service.

Claude can also help with work that lasts longer than one reply:

- **Research** gathers and combines information from several sources. You will learn about this in *Chapter 1.9 Conduct Research with Cited Sources*.
- **Projects** keep related chats, instructions, and files together. You will learn about this in *Chapter 1.11 Organize with Instructions, Memory, and Projects*.
- **Cowork** can complete longer assignments and produce documents, slide decks, spreadsheets, and reports. You will learn about this in *Book 2*.
- **Scheduled tasks** can run once, repeat, or watch for a change. You will learn about this and other automation features in *Book 3*.

A work platform gives you several ways to reach an outcome. You choose the goal, supply useful context, set limits, and review what comes back. Claude supplies speed and a wide base of general knowledge. You remain responsible for the purpose and the final decision.

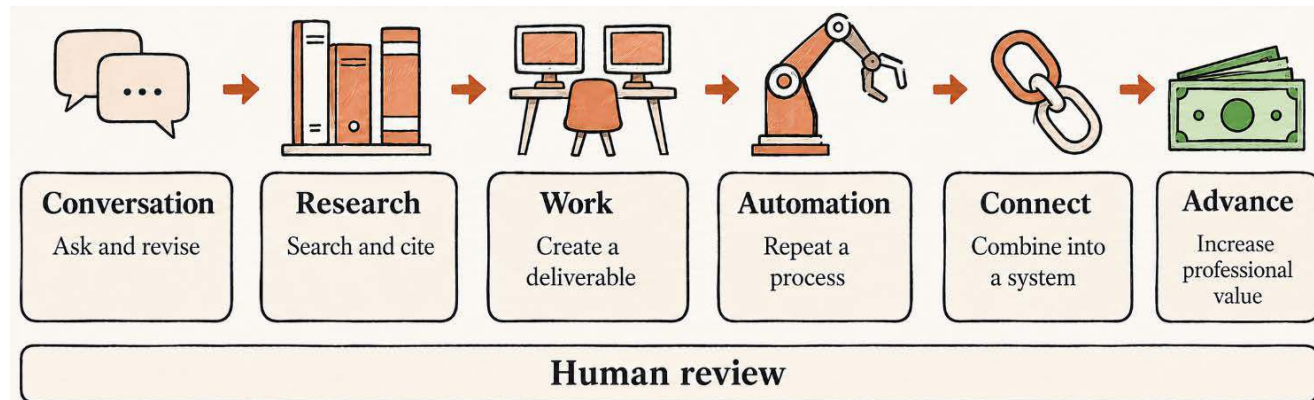


Figure 1.1A: A conversation can grow into checked research, finished work, a repeatable process, connected systems, and measurable value. Human review must span every stage

Tasks Claude handles well

Claude works best when you can describe the result and check the response. A suitable task has enough information, a clear audience, and a useful format. It also has consequences you can manage if the answer is incomplete or wrong:

Type of task	What Claude can help you do	What you should check
Explore	Suggest choices, questions, risks, or another approach.	Assumptions, bias, and options it left out.
Research	Search sources, compare claims, and prepare a cited report.	Source quality, dates, quotations, and claims.
Learn	Explain a topic, give examples, or make a quiz.	Accuracy, missing points, and whether the explanation fits you.
Analyze	Find patterns in a table and create a chart or explanation.	The source data, calculations, labels, and scale.
Create	Draft a message, plan, diagram, chart, document, or presentation.	Facts, fit for the audience, permissions, and final polish.
Transform	Summarize, rewrite, translate, classify, or extract details.	Lost meaning, tone, privacy, and copyright.

Table 1.1: Tasks that Claude handles well



Learn more online: *Claude Capabilities Overview*: <https://support.claude.com/en/articles/8114491-get-started-with-claude>.



Good practice: The best results come from pairing AI help with a check you know how to perform. If you do not have the experience to judge the output, ask a trusted person or qualified professional to review it.

Community arts center example

Imagine a community arts center wants more people to attend its weekend workshops. A staff member could upload survey data and approved planning notes, and then ask Claude about them:

Our weekend workshops have low turnout. Here's the survey data and planning notes – find patterns, check public research, compare three options, and draft a one-page recommendation.

Figure 1.1B shows Claude's response and annotates some common **Chat** mode user interface elements:

1. Select **Close sidebar** to make more space for your chat conversations.
2. Select the chat options button to pop up actions like **Star**, **Mark as unread**, **Rename**, **Add to project**, and **Delete**.
3. Files added to a chat, like Word documents and Excel spreadsheets, have their file types recognized automatically.
4. Select **Recents** in the sidebar to show and hide your recent chats.
5. Hover over **Recents** and select the switches icon to organize them into groups.
6. Select the down-arrow button to quickly scroll to the bottom of the response.
7. Select the + button in the chat box to add files to your chat prompt.

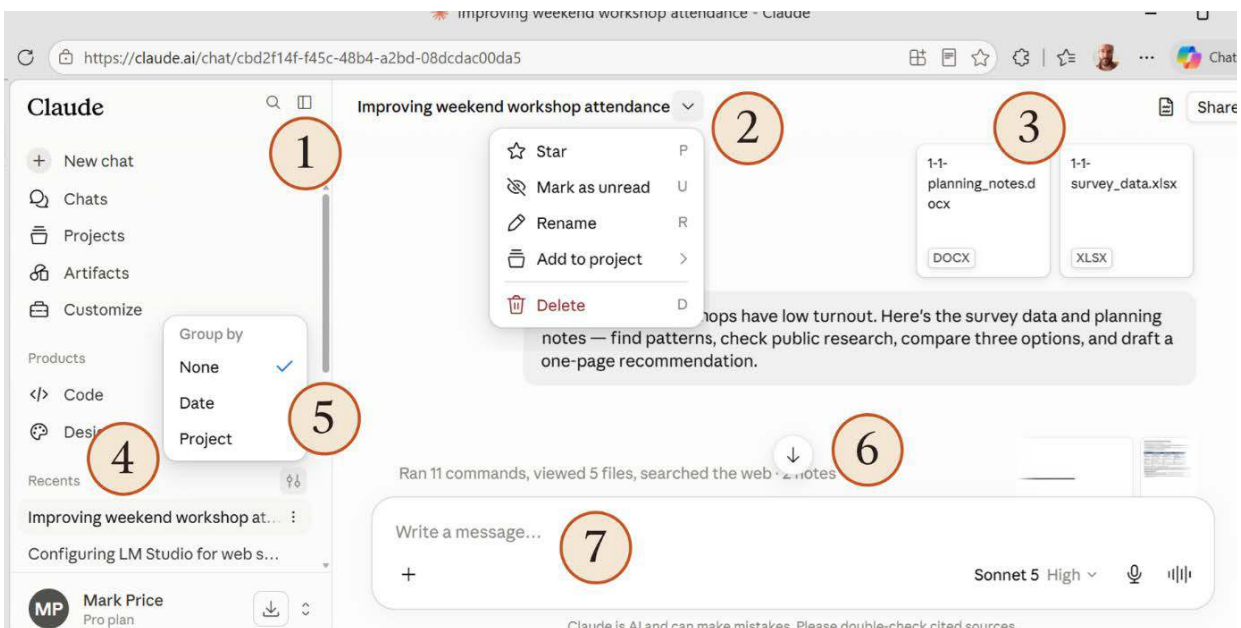


Figure 1.1B: Chat mode user interface elements

For this example prompt, Claude will compare three options, find patterns in public data, and draft a one-page recommendation:

8. Select the sidebar icon to open the sidebar when it's been closed.
9. Select the document in the response to open it in a sidebar on the right. This is called the **Artifacts** panel.
10. The response-generated document can be opened in Google Drive, downloaded, expanded to make more space, or closed with the x button.
11. The chat box has controls to change the model and effort, as well as record a dictated prompt or use voice mode.

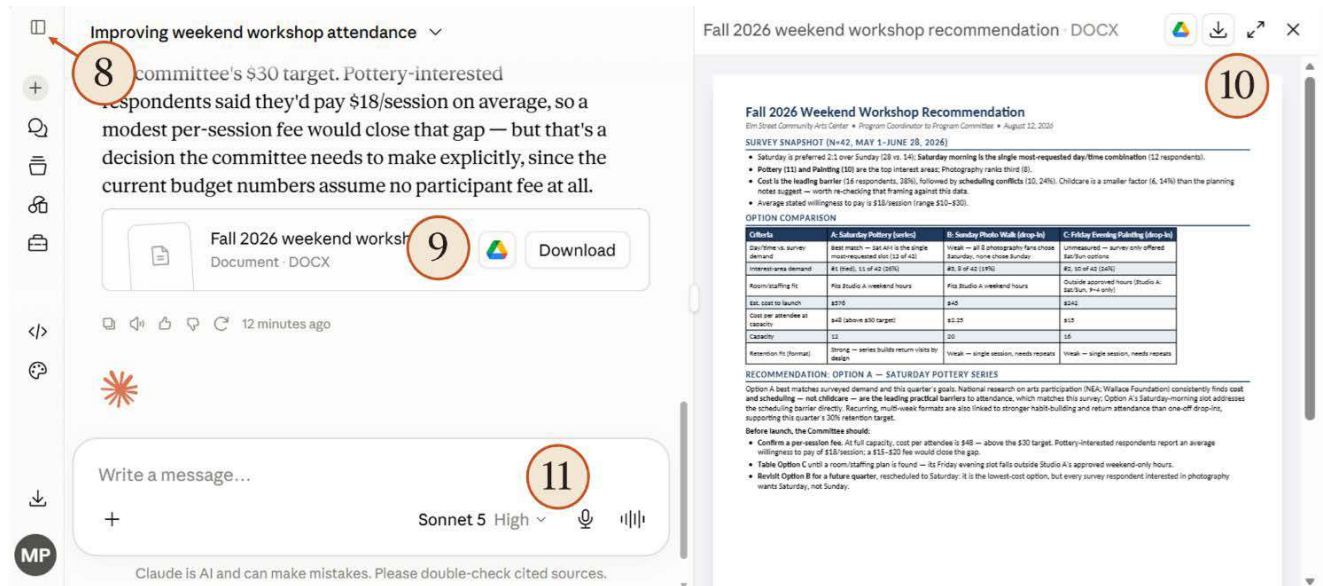


Figure 1.1C: The response includes a document in the artifacts panel that is ready for human review

The staff member would then check the data, open the sources, and decide which option fits the center's budget and goals.

Chat features

You can activate additional features by clicking the + button on the left side of the chat box:

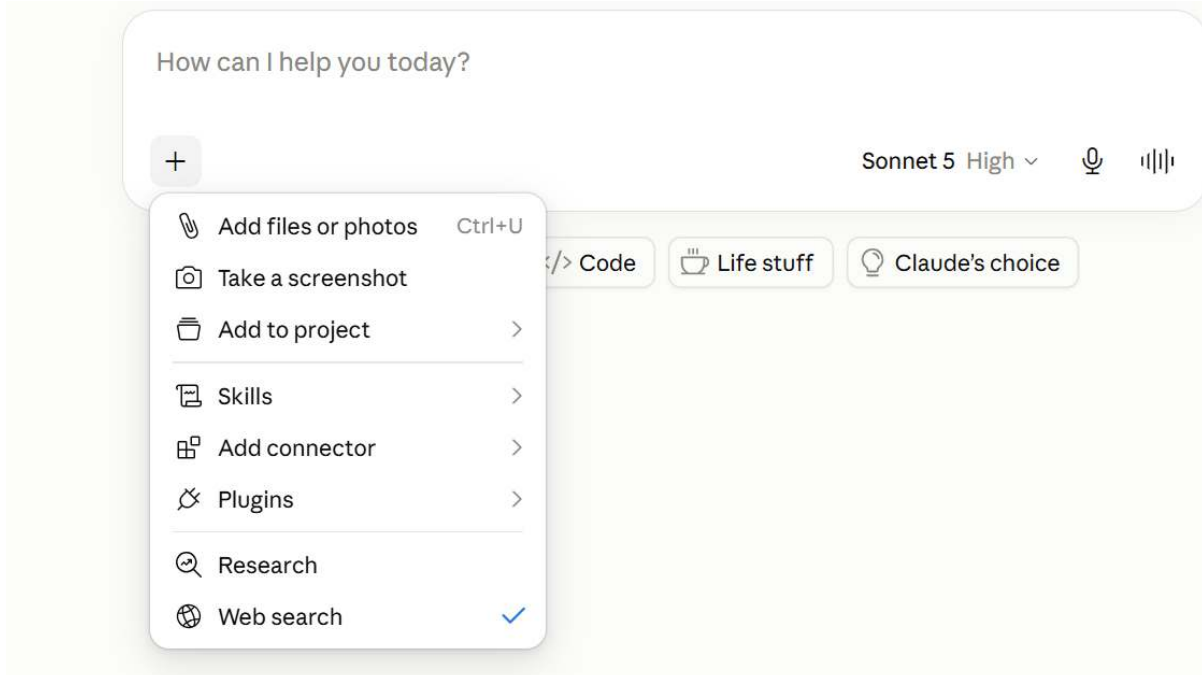


Figure 1.1D: Activating extra features in Chat mode

For example, if you want the response to be as up-to-date as possible with the latest information, make sure that **Web search** is enabled for this chat so Claude does not only rely on its model data that will have a cut-off date in the past.

You will learn about these extra features in the following chapters:

- **Add files or photos, and Take a screenshot:** *Chapter 1.10 Add Documents and Images to a Chat*
- **Add to project:** *Chapter 1.11 Organize with Instructions, Memory, and Projects*
- **Skills, Add connector, and Plugins:** *Chapter 2.8 Add Plugins, Connectors, and Skills*
- **Research:** *Chapter 1.9 Conduct Research with Cited Sources*
- **Web search:** *Chapter 1.8 Search and Verify Current Information*

Tasks that still require human judgment

Claude can produce a polished answer that contains an error. It may misunderstand your goal, omit a useful option, misread a file, or present an uncertain claim with too much confidence. Anthropic tells users to check important information and review output before using or sharing it.

It is especially important to keep a person in control when a task involves any of these conditions:

- **Facts with consequences:** Check names, dates, numbers, quotations, citations, and calculations against dependable sources.
- **Private or restricted information:** Do not upload passwords, private keys, health records, customer data, or confidential work unless you have permission and an approved account and process.
- **Professional advice:** Ask a qualified person to review medical, legal, financial, employment, or safety guidance before anyone acts on it.
- **Decisions about people:** Do not let Claude make the final choice about hiring, education, housing, credit, insurance, health care, or access to an essential service.
- **Actions that are hard to undo:** Review before a tool sends, publishes, buys, deletes, changes access, or contacts another person.
- **Values and relationships:** Claude can help you prepare for a choice or conversation. It cannot decide what you owe another person or which tradeoff you should accept.

Before you use an AI response, ask three questions:

1. **Is it accurate?** Check the facts and make sure the response answers the task you set.
2. **Am I allowed to use it?** Check privacy, consent, copyright, workplace rules, and access rights.
3. **What happens if it is wrong?** Add more review when the cost, harm, or loss would be hard to repair.



Watch out: Do not confuse a neat format with a correct result. A table, citation, or professional tone can make a weak answer look dependable. Open the sources and check the underlying facts.

From research to professional value

A long response does not create value by itself. Value appears when the work helps someone understand a problem, make a sound choice, create a useful result, or complete a task with less wasted effort. You need evidence of that outcome.

Let's return to the arts center example. A short chat might produce workshop ideas. Web search could add current evidence from the Internet. Adding a file for analysis could show which past workshops attracted repeat visitors. A verified recommendation could help staff choose one change. A reusable project could hold the source files and instructions for the next review. A scheduled task could check

enrollment each week.

The final step is to measure what changed. The center might track attendance, repeat bookings, staff time, or cost per booking. A professional using Claude could track turnaround time, errors found during review, customer response, or work completed. These measures show whether the process helped. They also show where it needs refinement.

Commercial value follows the same rule. A client or employer pays for a dependable result, not for the number of prompts you used. You earn trust by understanding the problem, protecting information, checking the work, and accepting responsibility for the deliverable.



Good practice: Start with a small, low-risk task that you already understand. Compare the time and quality with your usual method. Keep the parts that help, and change or remove the parts that do not.

Try it now

In this activity, Claude will help you decide where it can assist and where you should review. You'll work through this using one recurring task from your own work or personal life:

1. Choose one recurring task from your work or personal life. Pick a task that does not require private information and will not send, publish, buy, or delete anything.
2. Write down the name of your task in your notes. Examples include planning a weekly meal, preparing meeting notes, studying a topic, or comparing routine purchases.
3. Remove names, account details, confidential facts, and other information that you should not share.
4. Open Claude and paste the following prompt (if Claude asks you to sign in, save the prompt and return to it after you have read *Chapter 1.2* where you will learn how to register an account):

I want to improve how I handle [recurring task]. Ask me up to five short questions. Then create a table with these columns: current step, how Claude could help, what I must review, information I should not share, and one small test. Use only the facts I provide. Label every assumption. The first test must take less than 15 minutes and must not send, publish, buy, delete, or change access to anything.

5. Answer the questions without adding private or restricted information.
6. Read the table. Correct any false assumptions and ask Claude to revise the plan.



What to ask next: Choose the safest first test. Give me a short checklist, including what I should check before I use the result.

Check the result

- ☐ Does the plan separate Claude's help from the choice or approval that remains yours?
- ☐ Did Claude label its assumptions and identify information that you should not share?
- ☐ Can you test one part in less than 15 minutes without creating a hard-to-reverse consequence?

If any answer is no, tell Claude what is missing and request a new version. Revision is a normal part of working with an AI assistant.

If you have not created a Claude account yet, *Chapter 1.2* walks through that step and helps you choose a plan that fits the kind of task you just tried.

Chapter 1.2 Create Your Account and Choose a Plan

A free Claude account gives you saved chats, personal settings, and access on several devices. A paid plan can provide higher usage allowances and more access to some tools, but you do not need one to begin.

This chapter shows you how to create and protect an account, compare plans, recognize usage limits, and choose privacy settings you understand. Start free, then upgrade only when you can name the repeated need that a paid plan will solve.

Choose a sign-in method that you expect to keep. If an employer or school gives you a managed account, follow its rules and keep personal activity in a separate personal account. You will not buy a plan in this chapter.

Creating an account and checking your usage

You'll need to sign in to use Claude, but it takes seconds and keeps your chats and settings together across devices. It also lets you return to a chat on another device. Create a personal account for personal use, or use the managed account supplied by your organization for approved work.

The available sign-in choices can vary. You may be able to use an email address, or an existing account from another provider like Google that most people already have so you don't have to remember another password. Use a method that only you control and remember which choice you made. You will need the same method when you sign in later.

Create the account before you change its security settings:

1. Open your browser and enter `claude.ai` in the address bar. You will be automatically redirected to the login page. Check the website address spelling before you enter any account details:

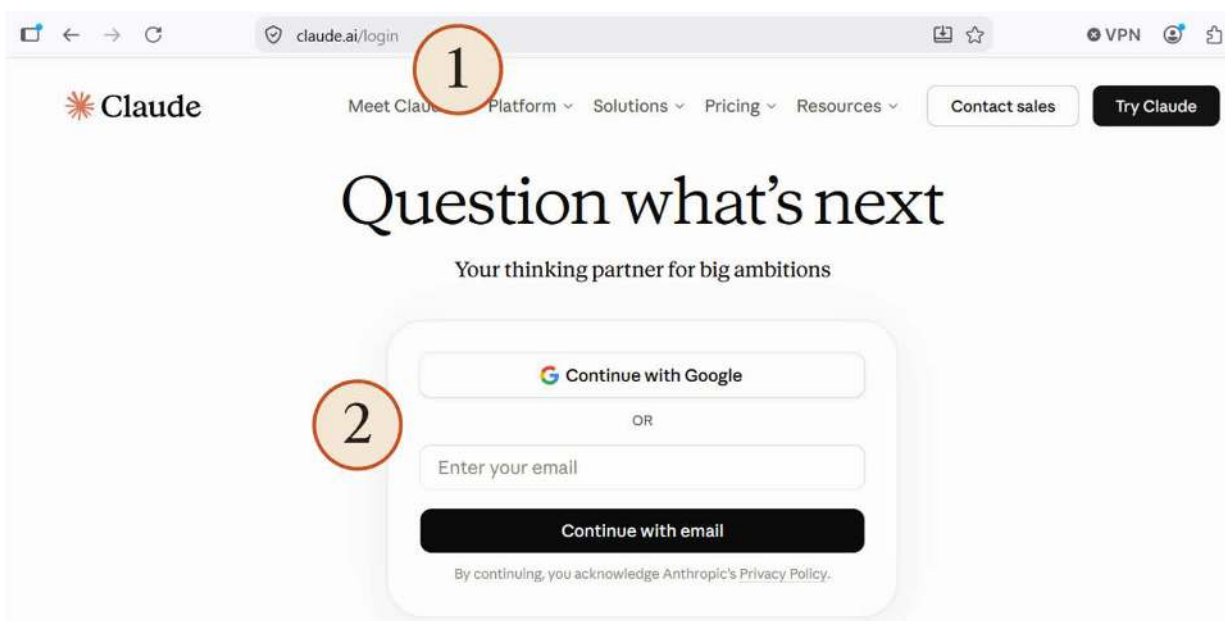


Figure 1.2A: Sign up for free to create your account on the official Claude AI website

2. Choose one of the sign-in methods shown on your screen. If you have a Google account, you can select **Continue with Google**. Or enter your email and select **Continue with email**.
3. Complete any identity or age checks that Claude requests.
4. Return to Claude after verification. You should now see a new chat and your account menu in the bottom-left corner of the Claude page.
5. Open your account menu by clicking on your name and plan. (For me that is **Mark Price** and **Pro plan** because I pay about US\$20 per month for the Pro plan.)
6. In the menu, select **Settings**, and then in the **Settings** window, select **Usage** (or directly select **Usage** in the account menu):

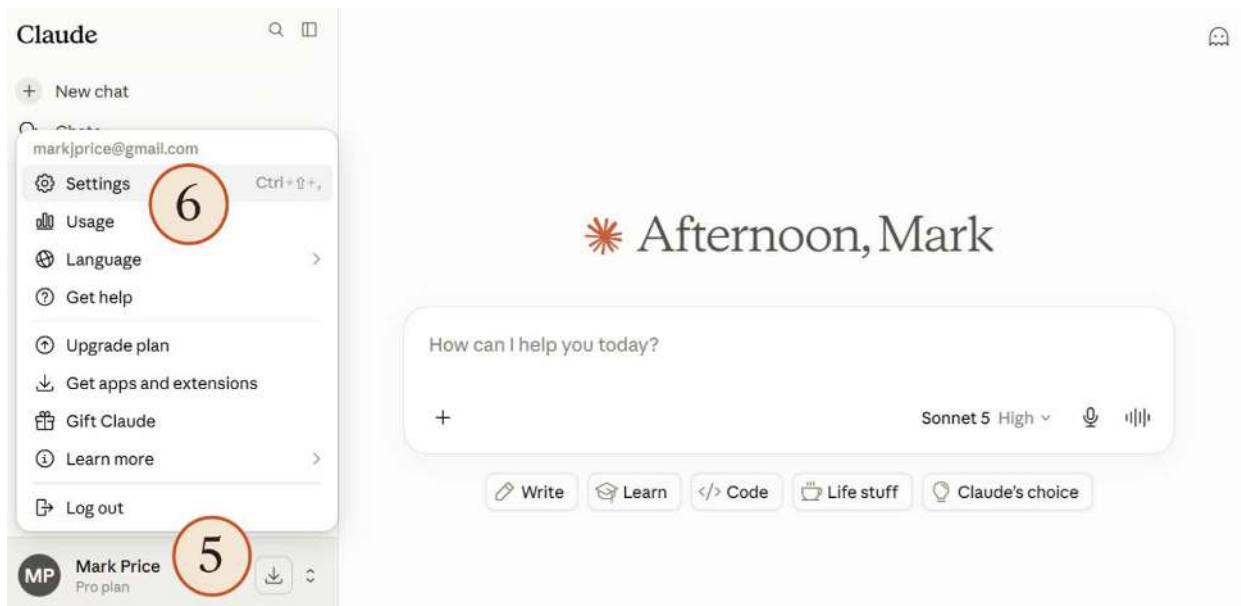


Figure 1.2B: Selecting your account menu and then Settings or Usage

7. In the **Settings** window, make sure you are in the **Usage** section.
8. Note your usage limits for the **Current session** (0% used in the screenshot shown in Figure 1.2C and it will reset in four-and-a-half hours) and for **All models** in the current week (9% used and it will reset to 0% on Sunday afternoon):

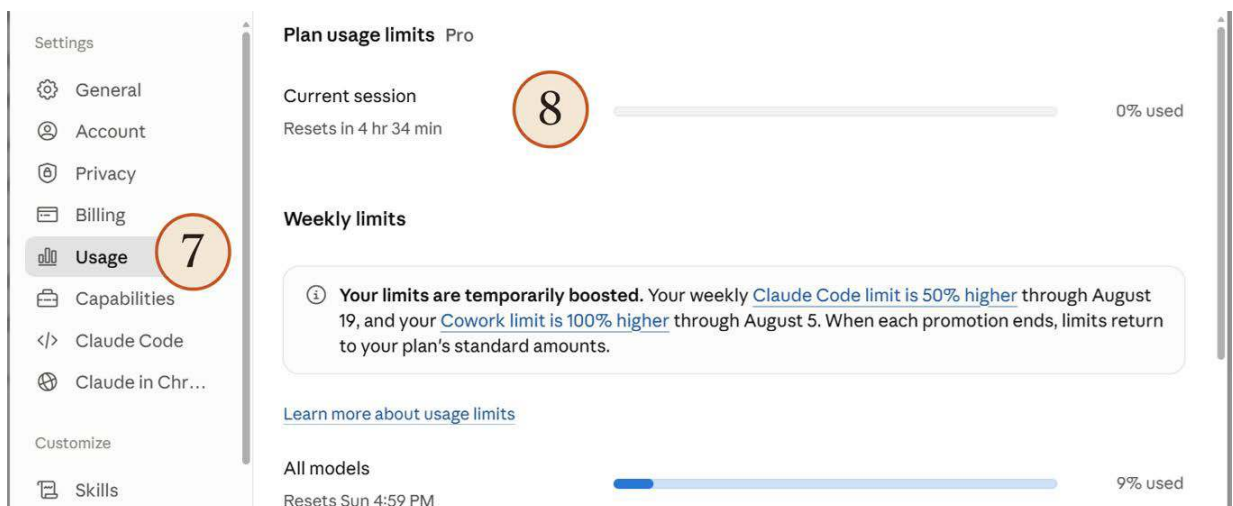


Figure 1.2C: Usage limits for the current session and the current week, and when they will reset



Good practice: Regularly check your usage while you work on heavy tasks to learn how quickly (or not) you use up your allowances during the current session or week. This will help you decide if or when you want to upgrade to a plan with higher limits.

Understanding usage limits and feature availability

A usage limit is a cap on how much of a model or tool you can use during a period. The **Free** plan has limits, and paid plans can have them too.

Different features can consume your usage allowance at different rates, and some capabilities also have their own file-size, file-count, or feature restrictions. Research, for example, uses the same overall conversation limits but can consume them faster because it searches and processes several sources. Read the message Claude shows you to learn which limit or restriction you have reached.

When Claude reports a limit, use this short check:

1. **Read the whole notice:** Look for the affected model or tool, the reset time, and any available alternative.
2. **Save your work:** If the limit will reset soon, continue later instead of buying a plan in a hurry.
3. **Record what the interruption prevented you from completing:** A repeated interruption gives you evidence for a plan decision.
4. **Open the current pricing page before upgrading:** Check the exact feature, allowance, billing period, tax, and cancellation terms shown for your country.



Learn more online: A tool shown in this book may appear in a different place, have a new name, or require another plan by the time you read it. Check the plan page and the notice inside your account: <https://claude.ai/upgrade/>

Understanding free and paid plans

Anthropic currently offers Free, Pro, and Max plans for individuals. **Team** and **Enterprise** plans are intended for organizations. Names, prices, features, and country availability can change, so check the current pricing page before paying.

Figure 1.2D shows how the **Free** plan will encourage you to **Upgrade**:



Figure 1.2D: The Free plan experience

The **Free** plan supports common chat tasks and includes limited access to tools such as web search, file uploads, and data analysis. An individual paid plan like **Pro** adds higher allowances, advanced models, and broader access to selected tools like Research. Max is aimed at people who use those capabilities even more often:

Plan	A suitable starting point	Question to ask
Free	You are learning, testing tasks, or using Claude from time to time.	Can I finish my usual task before a limit interrupts it?
Pro	You need higher allowances or a named feature for recurring personal work.	Which feature or allowance will I use each week?
Max	You use Claude often and regularly reach the limits of another individual plan.	Will the time saved justify the higher monthly cost?
Team or Enterprise	Your organization needs managed accounts, shared controls, or business data protections.	Which plan and settings does my organization approve?

Table 1.2: Choosing between free and paid plans

A paid plan does not make every answer correct. It also does not guarantee unlimited use of every model or tool. You still need to check facts, sources, files, calculations, and any output you plan to share.

Use a managed plan for work when your organization requires it. **Team** and **Enterprise** plans have different data protections and administrator controls. Ask who can access conversations, how long data is kept, which tools are enabled, and what information you may upload.



Good practice: Stay on the **Free** plan while you learn. Consider an upgrade to a paid plan after a limit or missing feature interrupts the same useful task more than once. Individual plans can be paid monthly, or annually at a small discount. The main benefit of monthly is that you can stop at any time instead of being committed for a full year.

Set the interface language

Separately from the language Claude replies in, you can change the language used for its own menus, buttons, and settings screens. Open your account menu, select **Language**, and then pick a specific language from the menu such as **English (United States)**, **Français (France)**, or **Español (España)**.



Watch out: This setting changes only the interface text. It does not change the language Claude uses to reply to you; that follows the language of your own messages unless you tell it otherwise.

Reviewing privacy and data controls

Privacy has several parts. Chat history, model improvement, memory, account administration, and data retention are separate controls.

1. **Model improvement:** In a personal account, open **Settings**, select **Privacy**, and review **Help improve our AI models**. Turning it off keeps new conversations in your history but prevents them from being used to improve Anthropic's models.

2. Export your data: Privacy also lets a signed-in user request a data export:

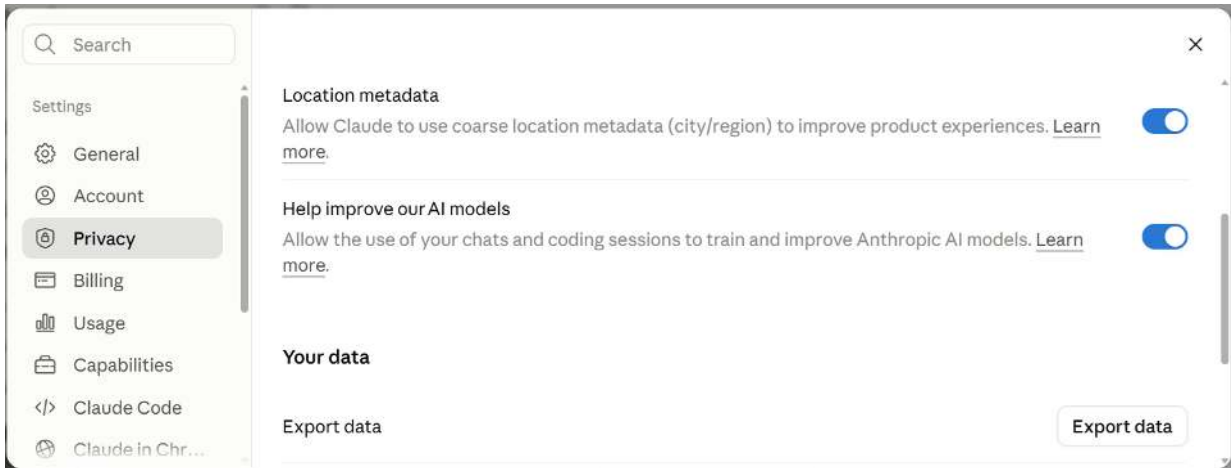


Figure 1.2E: Review model improvement separately from exporting data or deleting the account

Learn more online: *Export your Claude data:* <https://support.claude.com/en/articles/9450526-export-your-claude-data>.

Other privacy and data controls include:

- **Incognito chat:** Select the ghost button when you want to turn on a temporary chat when you do not want the conversation saved in history or used to create memories. Anthropic says incognito chats are not used for training and are deleted from its systems after 30 days, although they may be reviewed for abuse, and managed Team or Enterprise accounts may have different retention requirements, and organization administrators may be able to include these chats in data exports:



Figure 1.2F: Button in the top-right corner of Claude to turn on an incognito chat

- **Managed accounts:** An organization administrator may be able to access, export, audit, retain, or delete data in a managed account. The administrator can also restrict features and control retention settings.
- **Organization data:** Anthropic says it does not use content from **Team** and **Enterprise** offerings to improve its models by default. Your organization's own rules and administrator settings still apply.

No setting makes it suitable to paste passwords, private keys, confidential files, health records, or personal data that you have no right to share. Remove restricted information before you start and use an approved account for work.



Watch out: A managed work account is not private from the organization that controls it. Check that you are in the intended account before you begin a personal or work conversation.

Try it now

Create or review a free Claude account, add a second sign-in check, and record your plan and privacy choices. This activity should take 5 to 15 minutes. It does not require a purchase.

Open the Claude pricing page: <https://claude.ai/upgrade/> and compare **Free** with the next individual plan like **Pro** or **Max** available in your country. Do not purchase it during this activity.

Check the result

- ☐ Can you sign in again using a method you expect to keep, with another verification method enabled or its absence recorded?
- ☐ Did you compare the Free plan with the next available plan for your country without purchasing anything, and can you name one feature or allowance that would justify upgrading later?
- ☐ Did you review your privacy choices, and can you say what you chose for model improvement, and whether you know how to export your data if needed?

With your account created and secured, *Chapter 1.3* shows how to bring it with you: into a browser, onto your phone, and onto a desktop computer.

Chapter 1.3 Use Claude on the Web, Phone, and Desktop

Claude works in a web browser, on iPhone and Android phones, and in desktop apps for Mac and Windows. You can begin in one place and continue conversations in another, as long as you sign in to the same account.

This chapter helps you choose the version that suits the task in front of you. You will learn how to reach the official web and app downloads, how to check which account is active, and how to test a harmless conversation across two devices. You need one Claude account. A second device is optional.

The TL; DR is: Use the version that fits the task but check the active account before you write or attach anything.

Start with Claude in a web browser

The web version is the best starting point for most readers. It works without a separate installation and runs on any device with a modern browser, including locked-down work laptops, Chromebooks, and computers where you cannot install software. It gives you room to read a long answer, compare sources, and work with a document on a larger screen. Features can differ between the web, desktop, and mobile versions, and Anthropic changes them frequently.

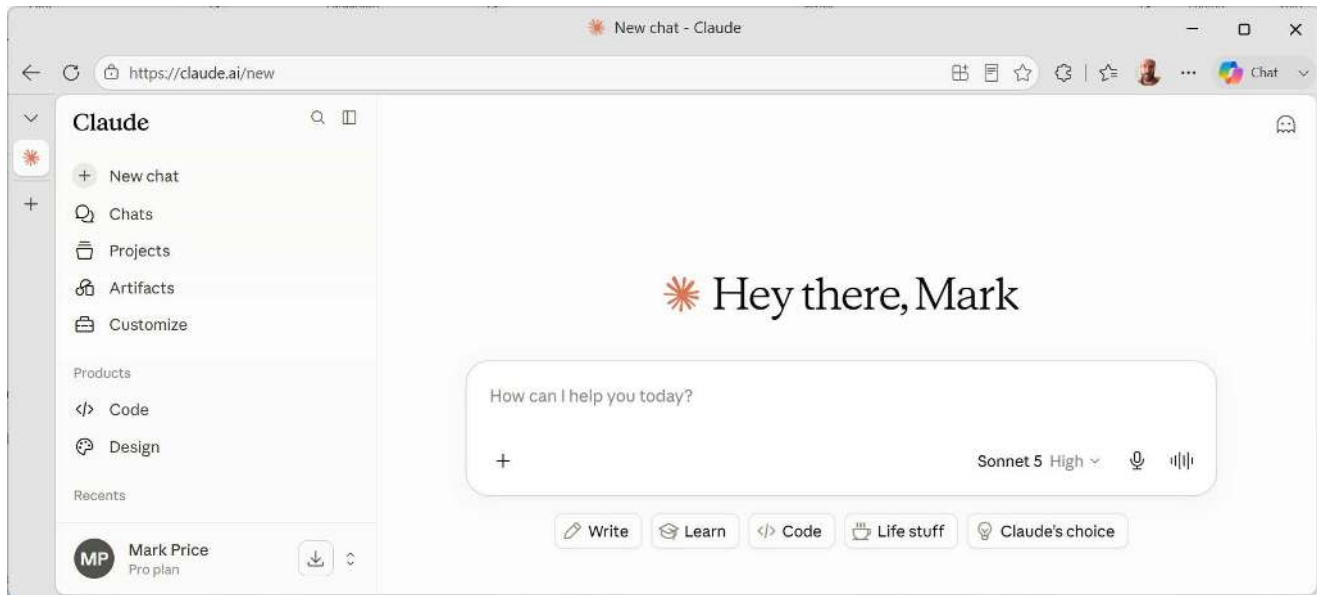


Figure 1.3A: Use the web version of Claude anywhere with a browser

Open a browser and enter `claude.ai` in the address bar. Then sign in with the same method you chose in *Chapter 1.2*. If you are using a shared computer, sign out when you finish and do not ask the browser to save your password.

A browser also gives you a useful safety check. Look at the address before you enter a password or upload a file. A search result, an advertisement, or a copied link can lead to a site that looks familiar but is not the official Claude service.

You can pin `claude.ai` in your browser or add it to your bookmarks. That makes it easier to return to the correct site without searching for it each time.



Watch out: In *Book 1*, we will only use **Chat** mode features. You will learn about **Cowork** mode in *Book 2*. Make sure you are in **Chat** mode for all the chapters in *Book 1*.

Use Claude on your phone

The official Claude app is available for iPhone and Android. The mobile app's strength is being useful away from a keyboard.

Voice mode lets you have a spoken, two-way conversation with Claude. On a phone, you can separately take or upload a photo and ask Claude about it, which is useful for objects, documents, signs, whiteboards, and other visual information.

This makes the mobile app best suited for quick, hands-free help while you're out and about rather than sitting at a computer:

1. Select the menu button in the top-left corner to toggle the sidebar.
2. The sidebar contains similar features to the web version. For example, **Recents** shows your recent chat conversations.
3. The chat box contains similar features to the web version, like **+** to activate extra features, a model and effort control, and dictate and voice mode buttons.

4. Select your account button (mine says **MP** for my initials) to view **Settings**.
5. **Settings** shows your profile, billing, and usage, as well as common UI options:

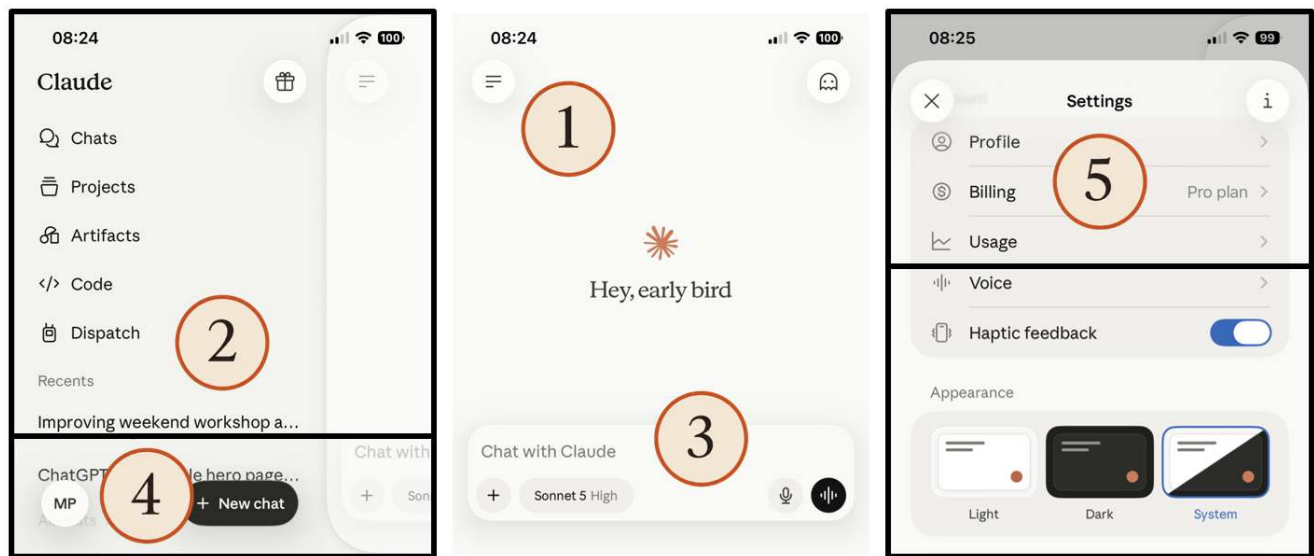


Figure 1.3B: Use the phone version on iPhone or Android devices

Download the iPhone app from the Apple App Store or the Android app from Google Play. Search for **Anthropic Claude** and check that **Anthropic** is the publisher before you install it. Country availability and device requirements can change, so use the store page if the app does not appear for you.

After installation, sign in with the same method as on the web. Sign in using the same Google account or email address that you use for Claude on the web. Using another sign-in method can create a separate account with a different chat history and plan.

The phone app may ask for permission to use the microphone, camera, photos, or notifications. Grant a permission only when you know why the feature needs it. For example, dictate and voice mode need microphone access and asking about a photo needs access to the photo you select.

A phone screen can hide context that would be obvious on a computer. Before you send a message, check whether you are in a personal account or a managed work account. Then reread any attachment name and the text above the send button.

Voice and photo features can capture more than you intend. Move to a private place before speaking about personal or work information. Crop or review a photo before you upload it, and remove names, addresses, account numbers, and other restricted details.

Use Claude on your desktop

The desktop app is intended for longer sessions at a Mac or Windows computer. It can keep Claude available while you work in a document, spreadsheet, browser, or other application. The exact features can differ from the web version and can change as Anthropic updates the app. The desktop app's advantage is system-level integration with whatever else you are working on.

Key user interface elements include:

1. The sidebar contains similar features to the web version.
2. You can filter recent chats by type (chat or task), status (active or archive), and last activity (1 day, 3 days, 7 days, 30 days, or all), and you can group them by date, type, unread, and status.

3. The desktop app has a toggle in the chat box to switch between **Chat** mode and **Cowork** mode. In *Book 1*, you will only use **Chat** mode. You will use **Cowork** mode in *Book 2*:

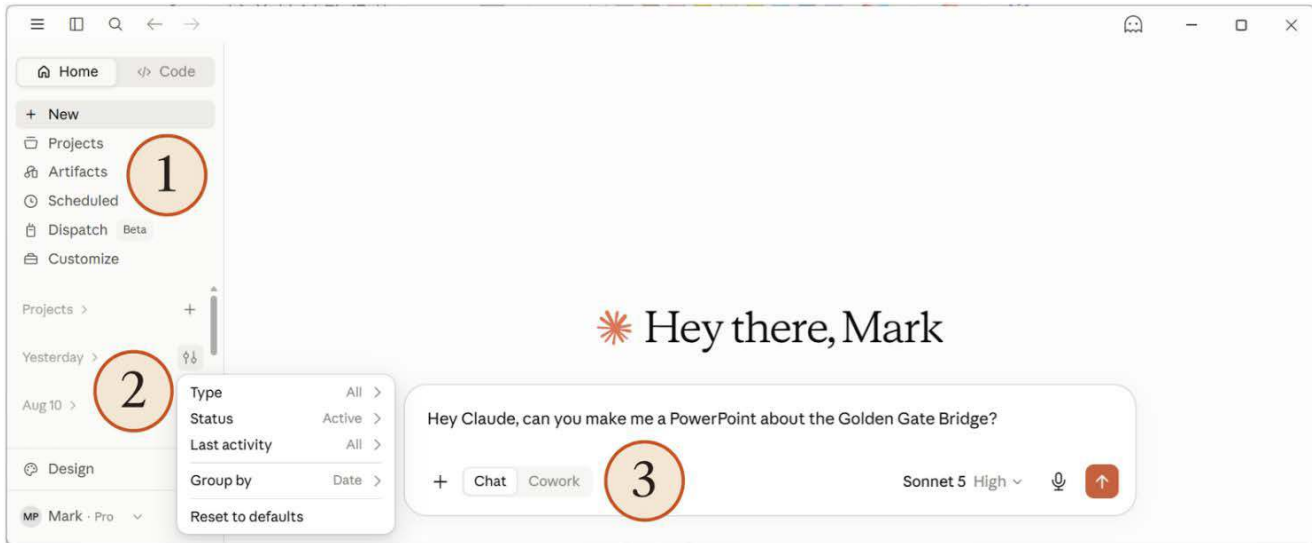


Figure 1.3C: Use the desktop version on Windows or Mac with a Chat and Cowork mode switch

Download the current desktop app from <https://claude.ai/download>, not from a search advertisement or a file-sharing site. Open the installer, follow the instructions for your operating system, and sign in to the same Claude account that you use on the web.

The current desktop app brings together Chat, Claude Cowork, and Claude Code. *Book 1* starts with ordinary Claude conversations. If you see Claude Cowork or Claude Code, you can leave those areas alone for now.



Watch out: Desktop features can request access to local files, other apps, a browser, or a work account. Read each permission screen. Do not give Claude access to a folder, service, or application because it might be useful later. Choose only the material needed for the task you are doing now.

If you use an employer's computer, ask whether the desktop app is approved before you install it. Your organization may manage app installation, restrict features, or require a work account for business information.



Learn more online: *Install Claude Desktop:* <https://support.claude.com/en/articles/10065433-install-claude-desktop>.

Open Claude instantly with a keyboard shortcut

You can quickly switch to the desktop app from whatever else you're doing with a keyboard shortcut. This is useful for a quick question while you're writing an email, reading a webpage, or working in another program.

The shortcut is off by default. Turn it on in **Settings | General | Quick Entry** keyboard shortcut, then select the clear button and press a key combination to set a new key combination:

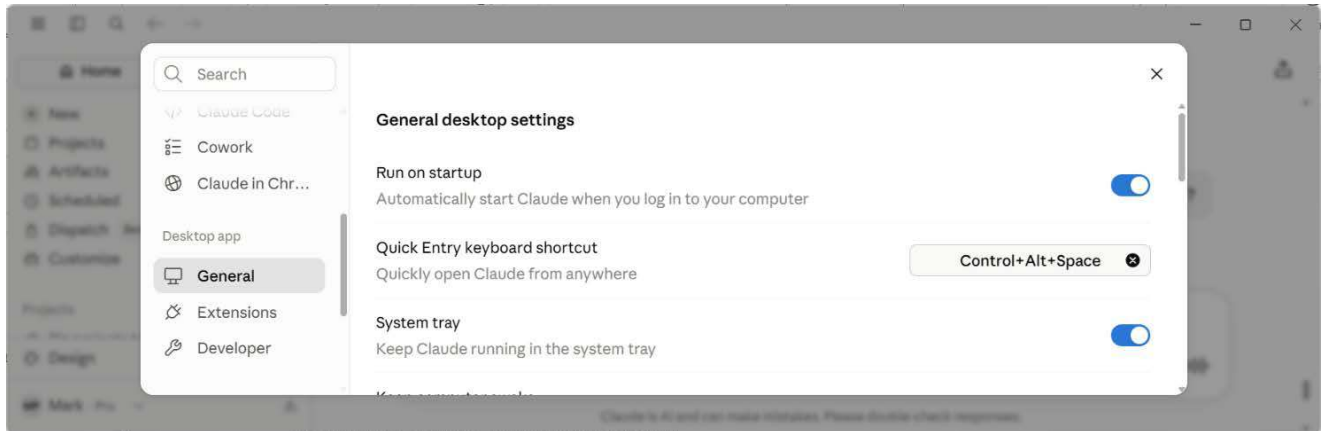


Figure 1.3D: Setting a `Ctrl+Alt+Space` key combination for the Quick Entry keyboard shortcut

Once it's on, press the shortcut from inside any other application to bring up the Quick Entry window:



Figure 1.3E: Quick Entry pop up window

Once you type a prompt and press *Enter*, it switches to the full Claude desktop app. If the shortcut doesn't respond, another Windows application may already use the same key combination. Choose a different combination in the same settings screen rather than assuming the feature is broken.



Good practice: Pick a combination you don't already use elsewhere. `Ctrl+Alt+Space` is a common choice because it avoids Windows' own reserved shortcuts. For Mac, try `Option+Space`, since it doesn't collide with Spotlight's built-in `Command+Space`, which in macOS 27 invokes Siri AI.

Continue a conversation on another device

Signed-in Claude conversations can usually appear on your other devices. That makes it practical to start a question in a browser, read the reply on a phone, and return to a computer when you need more space:

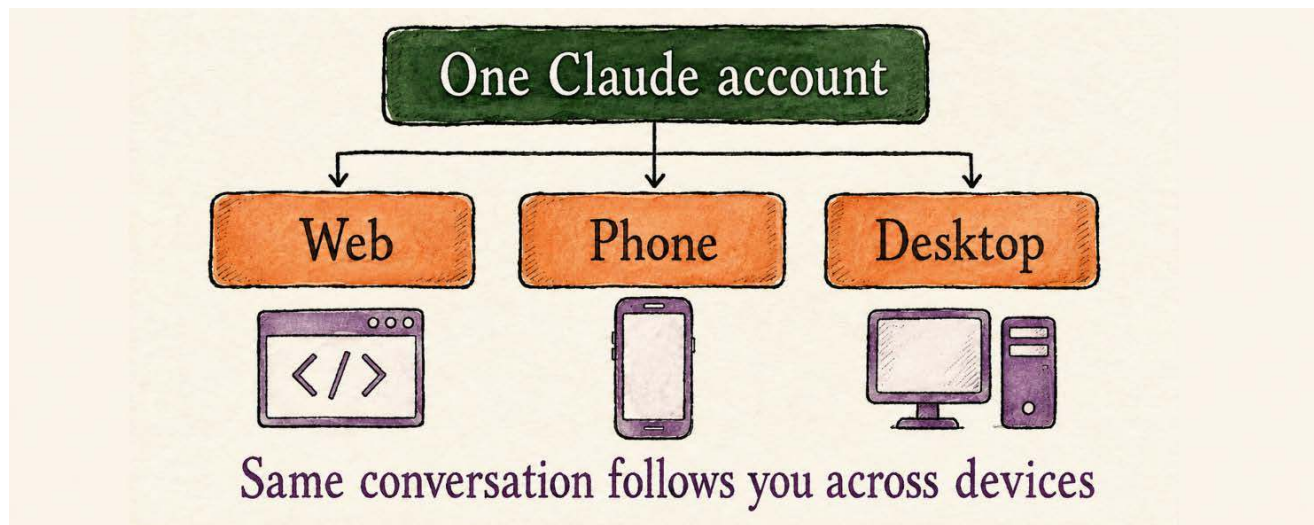


Figure 1.3F: Use the same account to continue your work across devices, while keeping temporary and managed-account conversations under the right controls

Use one conversation for one subject. If you switch from planning a vacation to writing a work memo, start a new chat. A focused conversation gives Claude less unrelated text to misunderstand and makes it easier for you to find the exchange later.

When a chat does not appear on another device, first confirm that both devices use the same account and workspace. Then check your internet connection, close and reopen the app or browser, and look for the conversation in chat history. Do not paste the same private material into a second chat until you know what happened to the first one.

Some activity stays tied to a particular device. For example, a task that uses local files or another installed app may need the original computer and your permission. Treat a new device as a new place to review the task before you continue it.

My recommendations for when to use each type of device and what to check before you begin are summarized in *Table 1.3*:

Device	Choose it when	Check before you begin
Web	You want a familiar browser window, a larger screen, or no installation.	The address is <code>claude.ai</code> and you are in the intended account.
Phone	You want to ask by voice, capture a photo, or continue a short task away from your desk.	The app is published by Anthropic , and the message contains no sensitive details.
Desktop	You work for long periods at a computer and want Claude close to your files and other apps.	Your computer and workplace rules allow the app and any permission it requests.

Table 1.3: Use Claude on the Web, Phone, and Desktop

Try it now

In this activity, you will sign in on the web and, if you have one, a second device. Then you'll confirm that a short test conversation appears in both places. This activity should take 5 to 15 minutes:

1. On a computer or phone browser, open `claude.ai` and sign in to the account you chose in *Chapter 1.2*.
2. If you have a second device, install the official Claude app from the Apple App Store, Google Play, or `claude.ai/download`. Check that Anthropic publishes the app or download page.
3. Sign in on the second device using the same sign-in method. Check the account name or workspace before you continue.
4. On the first device, start a new chat and send the following prompt:

```
Reply with the words: This is my cross-device test. Do not search the web or
use another tool.
```

5. Open Claude on the second device. Find the test chat in the chat list or search for the text: `cross-device test`. Read the reply, then add this follow-up:

```
I can see this conversation on another device.
```

6. Return to the first device and check for the follow-up. If it does not appear, confirm the account and workspace on both devices before you try again.



What to ask next: Help me choose whether I should use Claude on the web, phone, or desktop for these tasks: [list up to three tasks]. For each one, name the best starting place, one privacy check, and one reason I might switch devices.

Check the result

- ☐ Can you reach the official Claude website and identify an official mobile or desktop download source?
- ☐ Can you tell which account or workspace is active before you start a conversation?
- ☐ Did the harmless test chat appear on both devices, or did you record the point where it did not?
- ☐ Can you name one task that suits the web, phone, or desktop version and one privacy check for that task?

If an answer is no, use the web version as your starting point and check the current documentation for your device. If you use a managed account, ask your organization administrator before you install software or change access settings.

Once you know where to sign in, *Chapter 1.4* turns to what you do once you're there: having a first conversation you can type, speak, or show.

Chapter 1.4 Have Your First Useful Conversation by Typing, Talking, or Showing

A useful Claude conversation has a goal that you can describe and an answer that you can improve. You can type a message, speak through voice mode, or show Claude a photo or screenshot. The input method changes how you give information. The same conversation habits still apply.

In this chapter, you will make one small request, answer follow-up questions, correct a wrong assumption, and ask for another option. You will also learn how to keep a conversation that you may want to return to or share after review.

Languages Claude can understand and reply in

You don't need to change any setting to talk with Claude in your own language. Write or speak in the language you're most comfortable with, and Claude usually replies in that same language. Anthropic says Claude can work with most world languages that use standard text characters, and elsewhere says it can hold conversations in more than 200 languages. Performance varies by language and is generally strongest in widely spoken languages, so check anything important, such as a translation or technical term, just as you would any other answer.



Good practice: If you want Claude to answer in a specific language regardless of the language you type in, say so directly, for example, “Reply in Spanish even if I write to you in English.”

Choose the best way to start

Type or dictate when the wording or details need care. A typed or dictated request is useful for a question with names, dates, limits, or a format that you want to keep. You can reread it before you send it and edit it later.

Use voice mode to talk to Claude interactively. This will help you explain an idea or ask short follow-up questions. Use voice mode only if it appears in your account, then allow microphone access when you are ready. Voice transcripts can differ from what you said, especially with background sound or overlapping speech, so review the transcribed text before you rely on it.



Watch out: Do not confuse the Dictate feature (the microphone button) with using voice mode (the waveform button). Dictate mode is an alternative way of typing. You are still ultimately entering text.

Show Claude an image when the picture contains information that words would describe poorly. For example, you might show a non-identifying photo of a household object and ask what features it can see. Crop the image first and remove names, addresses, faces, account numbers, and other information that you should not share.

To send that image, or anything else you want to ask, use the controls in the chat box:

1. The + button lets you add a file like an image or photo and activate features like research.
2. Select the chat box to type text.
3. Press and hold the microphone button to record a dictated prompt. Release to stop recording.

4. Select the waveform button to use voice mode, and then speak to Claude naturally and more interactively. Select it again to stop:

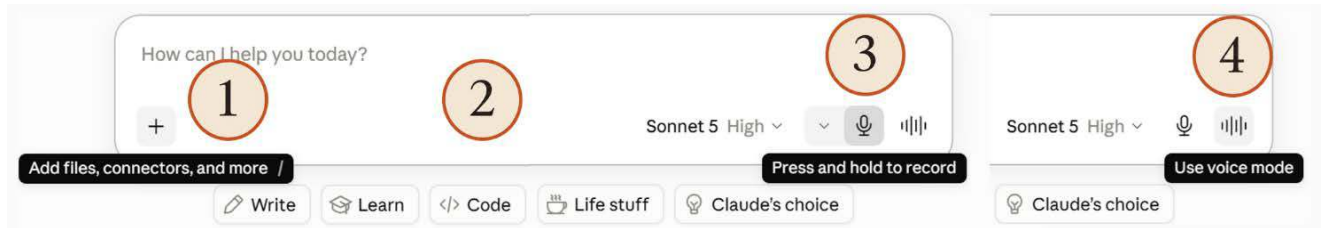
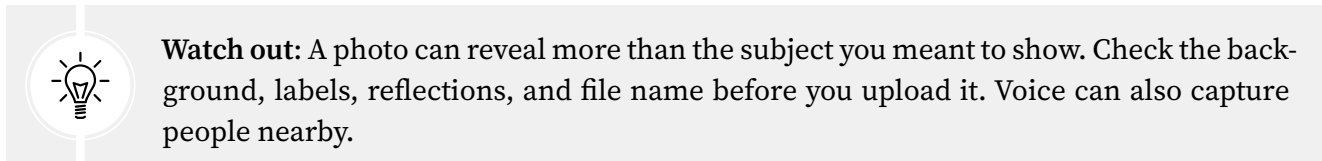


Figure 1.4A: Multiple inputs: add a file (like a photo you've taken), type or dictate text, and use voice mode



Select a model and effort level

Once you know how you will send your message, you can also choose how much effort Claude puts into answering it. A control in the chat box shows your current choice, such as **Sonnet 5 Medium**. Select it to open the model and effort menu, where you trade speed for depth, or depth for speed, before you send a request.

In an ordinary chat:

1. The model and effort control usually offers four models: **Fable**, **Opus**, **Sonnet**, and **Haiku**.
2. The **Effort** menu offers a number of levels, depending on your plan: **Low**, **Medium**, **High**, and, on some plans, **Extra** and **Max**.
3. The **More models** menu offers some older models like **Opus 4.8**, **Opus 3**, and **Sonnet 4.6**. Most people can leave this on the newest model. Switch only if you have a specific reason to keep using an older version, since Anthropic adds and retires versions in this list over time:

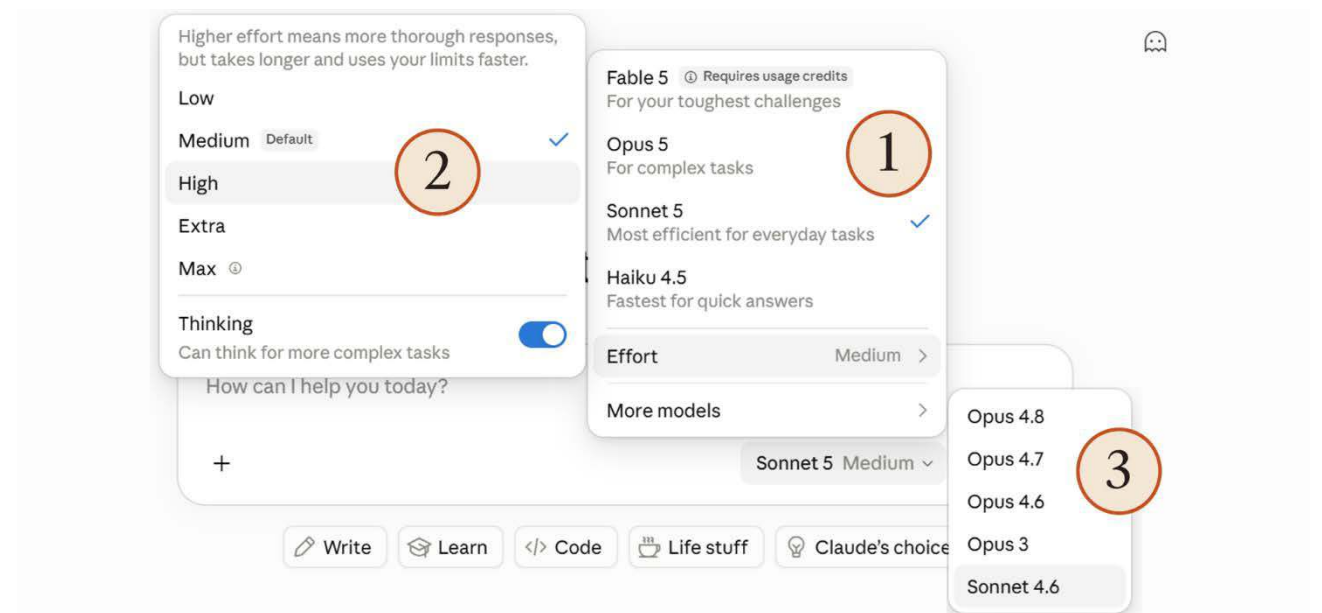


Figure 1.4B: Selecting models and effort levels

Each one balances speed against how much the model reasons before it replies:

Option	What it does
Low	Fast answers for everyday questions.
Medium	Standard reasoning for tasks that need some thought. This is the default for the Sonnet model in Chat mode.
High	Extended reasoning for harder tasks, such as planning or debugging. This is the default for the Opus model in Chat mode.
Extra	The most reasoning effort available in Chat mode.
Max	The highest-capability option, for difficult or long-running work.

Table 1.4: Levels in the Effort menu



Learn more online: *Change the model, effort, and thinking settings:* <https://support.claude.com/en/articles/8664678-change-the-model-effort-and-thinking-settings>.

Which of these options you see depends on your plan. Anthropic rolls out new model families across accounts over time, so check your own model and effort control for the exact list your plan currently offers.

A higher effort level does not guarantee a correct answer. It only gives Claude more brain power to work through the problem. But over-thinking is a genuine problem for AI just as it is for humans. Keep checking facts, sources, and calculations at every level, the same way you would with Low.



Good practice: Match the effort level to the task. Use Low for a quick fact or a short everyday message. Move up to Medium or High when a task involves several steps, code, or a decision worth checking carefully. Save Extra or Max for work that is genuinely difficult or long, since these levels take longer to answer and use more of your plan's usage allowance.

Ask for a specific result

Start with the result you want, who it is for, and any limit that affects the answer. Then ask for a format that helps you compare or use the response. You do not need a long instruction. The first request gives Claude enough information to make a useful start.

For example, this request names the task, the limit, the result, and a check before a purchase:

```
I am choosing a gift for a neighbor who enjoys gardening. Ask me up to three
questions. Then suggest three gifts that cost no more than [amount and currency].
Give one reason for each choice and name anything I should check before buying. Do
not make a purchase.
```

The request does not assume that Claude knows the neighbor, local stores, or your budget. It asks for questions first. That gives you a chance to add information before Claude makes suggestions. This example illustrates two things about prompting well:

1. Asking a broad question will not give you a great response
2. Providing some detail starts a conversation that can get you what you need

Answer questions, correct assumptions, and fill gaps

Claude may ask a question before it answers, or it may make an assumption that does not fit. Treat either response as a chance to guide the work. Answer the useful questions. State what is wrong. Add the missing fact. Then ask for a revised answer.

For the gift example, you might say:

```
The neighbor has a small balcony, not a garden. They already own basic tools. Revise the suggestions. Keep the budget below [amount and currency], and tell me which option needs no special equipment.
```

A correction does not need to restate the whole task. It only needs to change the part that was wrong. If Claude still guesses, ask it to label each assumption and list the information it needs from you.



Good practice: When a response contains a fact that could affect money, health, safety, work, or another person, check the fact before acting. A useful conversation can still contain an error.

Request alternatives and make your choice

One answer may be suitable, but it is rarely the only possible answer. Ask for alternatives when you need to compare cost, time, risk, style, or another tradeoff. A short table makes the differences easier to inspect.

For example, you could ask:

```
Give me three different options. Put them in a table with these columns: option, estimated cost, why it fits, possible drawback, and what I need to check. Do not invent facts that I have not supplied.
```

Read the result as a set of ideas, not as a final decision. You choose the option, check the details, and decide whether the answer is ready to use. If none of the options work, say why and ask for another set.

Requesting alternatives and making your choice follows a five-step loop:

1. Name the result
2. Answer questions
3. Correct assumptions
4. Compare alternatives
5. Review or keep useful work

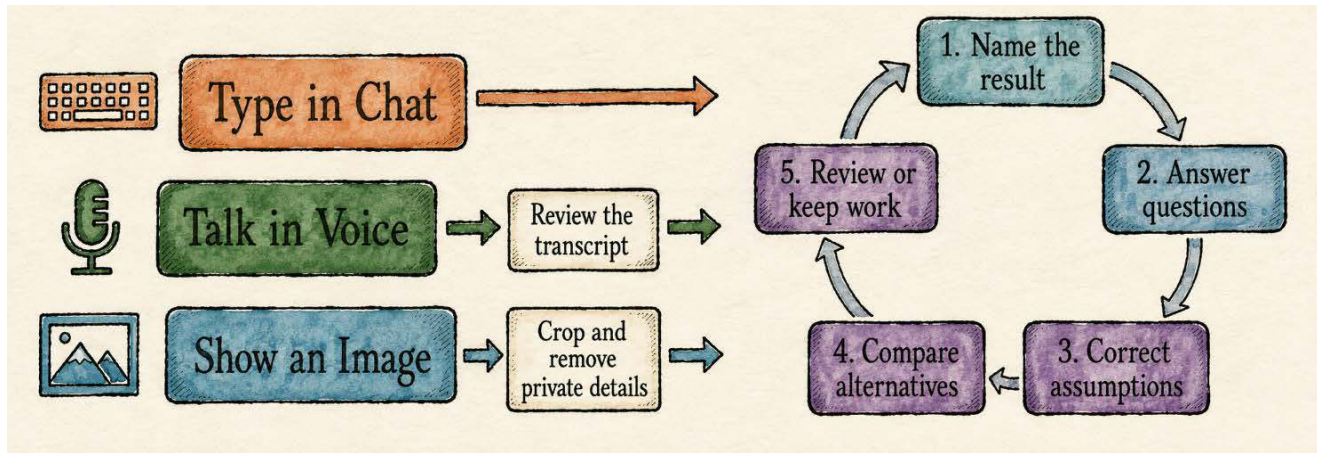


Figure 1.4C: The input can change. The improvement loop stays the same.

For voice input, always review the transcript to ensure that Claude correctly understood your request. For image input, crop and remove any private details before sending the image to Claude.

Human review of the response and comparison with alternatives before using it is always good practice.

Keep useful work without sharing too much

A useful conversation is easier to find when its title describes the work. Claude may give a chat an automatic title. Open the chat options and choose **Rename** if that control appears, then use a title such as “Balcony gift ideas, July 2026.” The exact location of chat options can change across devices.

Copy a finished result into your own document when you need to keep it with other work. Include the sources, choices, and checks that support the final decision. A chat is a working record. Your document can become the version that you edit, approve, or share.

Claude can create a shared link for a conversation in supported accounts. Anyone who has the link can view the conversation and pass it on. Read the full conversation before you share it. Remove private, confidential, or unnecessary material. Use a managed work account and your organization’s rules for work conversations.

Try it now

Hold a short conversation that improves over several messages. This activity should take 5 to 15 minutes:

1. **Start a new chat:** Choose a small planning task, such as a 60-minute activity for a community group, a meal idea, or a gift for someone you know. Use typing, Voice, or a non-identifying photo. Do not upload a file, share a link, or make a purchase.
2. **Send a first request that names the result you want:** Ask Claude to ask up to three questions before it suggests an answer.
3. **Answer the questions:** If you used Voice, read the transcript and correct any words that were misunderstood. If you used an image, check that Claude describes only what you can see.
4. **Send one correction or missing detail:** Ask Claude to revise its answer and label any remaining assumptions.
5. **Ask for three alternatives in a table:** Name one fact, cost, source, or safety point that you would check before using an option.

6. **Rename the chat so that you can find it again:** Do not share the conversation link during this activity.



What to ask next: Review this conversation. List the facts I need to verify, the assumptions you made, and the private information I should remove before sharing any part of it.

Check the result

- ☐ Did your first message describe a result, a limit, or a format that you wanted?
- ☐ Did you answer a useful question or correct one assumption before asking for a revision?
- ☐ Can you explain one check that you would make before acting on the response?
- ☐ Did you rename the chat without sharing private or work-restricted information?

If an answer is no, return to the chat and add the missing information.

Chapter 1.5 breaks that same instruction into four parts you can reuse for almost any task: the outcome, the context, the limits, and the form of the answer, and shows what to do once Claude's first answer is close, but not quite right yet.

Chapter 1.5 Write, Refine, and Improve Prompts Through Conversation

A prompt is the message you give Claude. A useful prompt tells Claude what you want to do and what a good answer should include. You do not need special wording or a long list of rules. You need to provide enough information in everyday language for Claude to make a suitable first response.

In this chapter, you will write one prompt in four parts: the task and outcome, useful background, limits, and the form of the answer. The same pattern works for planning, learning, writing, and many other everyday tasks.

State the task and desired outcome

Start by naming the job you want Claude to help with. Then say what you want at the end. A task is an action, such as plan, explain, compare, draft, or list. An outcome is the result you can inspect, such as an agenda, a checklist, a table, or a short draft.

For example, imagine that you are helping a community group run a 60-minute class for adults who are new to smartphone photography. "Help with a class" leaves many choices open. "Create a 60-minute class plan" gives Claude a result to work toward, like so:

```
Create a 60-minute class plan that helps adults who are new to smartphone photography take, find, and share photos safely.
```

The first sentence is enough to begin. You can add the details that affect the answer next. Do not try to predict every question Claude might ask. Include the information that would change the plan if it were missing.

Supply relevant context

Context is the background that helps Claude understand the situation. It can include the audience, what they already know, the purpose, available materials, and facts that shape the task. Give context that is relevant to the answer. Extra details can distract from the main job.

For the smartphone class, the group size, the kinds of phones people have, and the teaching location all affect the plan. A favorite movie or an unrelated work deadline would not help with this request.

Here is more context for the example prompt:

The class is for eight adults at a community center. Some use iPhones and some use Android phones. They can use their own phones, a projector, and a printed handout. Most have taken photos before but do not know where the photos are stored or how to share one.

You can write the task and context as a few short sentences. If the context comes from a document or image, check that you have the right to share it. Remove names, contact details, account numbers, faces, and other information that the task does not need.

Figure 1.5A shows:

1. **Form of the answer:** “Create a 60-minute class plan...” sets the length and the type of result. Claude returns a timed plan instead of general advice about photography.
2. **Task and desired outcome:** “...that helps adults who are new to smartphone photography take, find, and share photos safely” states the outcome you want. Claude can aim at a clear goal instead of a general topic.
3. **Constraints and boundaries:** “They can use their own phones, a projector, and a printed handout” lists what is available in the room. Claude can suggest activities that fit the space instead of ones that need equipment nobody has.
4. **Relevant context:** “The class is for eight adults at a community center. Some use iPhones and some use Android phones. Most have taken photos before but do not know where the photos are stored or how to share one.” This describes the learners, including group size, devices, and what they already know. Claude can match the plan to this group.

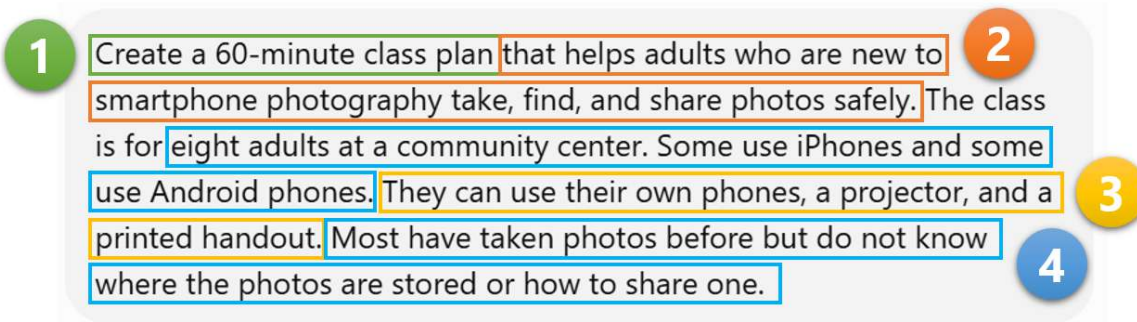


Figure 1.5A: Four kinds of information can make a first prompt more useful. Add only the parts that help with your task.

Set constraints and boundaries

Constraints are limits that the answer must follow. They can include time, budget, length, tools, language, or the number of choices. Boundaries say what Claude should not do. They are useful when a response could otherwise make an unsafe assumption, spend money, or use information you have not supplied.



Watch out: A boundary only rules out what you actually name. Telling Claude “do not invent local rules” does not stop it from inventing a different detail you forgot to rule out. State the boundaries that matter most, and still check the answer.

For the class plan, the group has only 60 minutes and must use the standard photo tools already on the phones. No participant should have to share a private photo. Those details help Claude choose suitable activities:

```
Keep the plan within 60 minutes. Use only standard phone features, not paid apps or extra equipment. Include an activity that works for both iPhone and Android users. Do not ask participants to share personal photos. Do not invent local rules or phone settings. Mark anything that I need to check on my own phone.
```

A boundary is not a promise that Claude will always get the details right. Read the answer and check any fact that affects privacy, money, safety, work, or another person. For an important decision, use qualified human advice where needed.



Good practice: Keep constraints and boundaries specific enough to check. A vague limit such as “keep it reasonable” cannot be verified. A specific one, such as “under 500 words” or “do not invent a fact,” gives you something to confirm in the response.

Specify the form of the answer

The form is how you want the answer arranged. Ask for a table when you need to compare items. Ask for a numbered list when you need actions in order. Ask for headings when you want a short guide. You can also set a length, a reading level, or a tone if those choices will help the reader use the result.

For a class plan, a table can make the timing and materials easier to check. You can name the columns rather than hoping that Claude chooses the same structure you had in mind:

```
Give the answer as a table with these columns: time, activity, what participants do, materials, and privacy reminder. Then add a five-item handout checklist in plain language. Keep the whole answer under 500 words.
```

Put the four parts together in one message when you are ready to send it. The order is less important than the information. If Claude asks a useful question, answer it. Claude should respond with a table containing a 60-minute class plan:

Smartphone photography class plan for beginners
Share

mascara, and privacy reminder: not ask a new item without checking in plain language. Keep the whole answer under 500 words.

60-Minute Class Plan: Smartphone Photography Basics

Time	Activity	What participants do	Materials	Privacy reminder
0:00-0:05	Welcome & goals	Listen; share one thing they hope to learn	None	—
0:05-0:15	Taking a good photo	Practice opening the camera app and taking 2-3 photos of	Own phones	Photograph objects, not other participants, without asking first

Figure 1.5B: A named result, useful background, clear limits, and a requested format make the response easier to review

A first answer gives you something to inspect, but it may still be missing information, make a poor tradeoff, or need one section changed. The rest of this chapter continues the smartphone-photography class example, asking Claude what it needs to know, comparing alternatives, checking an answer against your own criteria, and revising one part while keeping the parts you want to keep.

Handout: 5-Step Photo Checklist

1. **Take it** — Open your Camera app and tap the round button.
2. **Find it** — Open your Photos app to see the picture you just took.
3. **Choose it** — Tap and hold, or tap "Select," to pick a photo.
4. **Share it** — Tap the Share icon (a box with an arrow) and choose how to send it.
5. **Think first** — Only share photos you're comfortable with others keeping or forwarding.

Note for facilitator: Menu names and exact settings (Share Sheet layout, AirDrop location, Nearby Share toggle) vary by phone model and OS version — confirm these on the specific phones in the room before class.

Figure 1.5C: The class plan ends with a five-item handout checklist in plain language, as requested

Ask Claude to identify missing information

Claude can answer a question even when important details are missing. That can be useful for a rough draft, but it can also lead to guesses. Before asking for a larger answer, ask which missing facts would change the result.

Let's continue the smartphone-photography class example. The first plan may fit the time limit and use standard phone features. It still does not say whether everyone can see a projected screen, whether the group needs accessible materials, or how much help each participant needs. Those details may change the activities.

1. Request for missing information:

Before you improve the 60-minute smartphone-photography class plan, list up to five missing facts that could change the plan. Group them as audience, room and equipment, accessibility, and safety or privacy. Do not guess the answers. For each fact, say how it could change the plan.

2. Selective answers: Read the questions before you answer them. Some will be useful. Others may not affect your task. Answer only the ones that change your decision.
3. Labeled assumptions: If you do not know the answer, say so and ask Claude to make any labeled assumptions that you can review later.

Request alternatives rather than accepting the first answer

A response can meet your instructions and still be one of several suitable choices. Ask for alternatives when you need to compare time, cost, order, level of detail, or another tradeoff. Name what must stay the same, so Claude does not discard useful work.

For the class plan, one version might spend most of the time practicing with phones. Another might include more time finding and organizing photos. Both can fit the same audience and privacy boundary. Comparing them helps you choose the version that suits the group.

Here is an example prompt:

Give me three alternative 60-minute class plans. Keep the audience, standard-phone-features rule, and privacy boundary from the current plan. Make the main emphasis different in each version: taking photos, finding and organizing photos, or sharing photos safely. Put the options in a table with these columns: emphasis, main activities, advantage, possible drawback, and what I need to check.

Ask for alternatives that differ in a named way. A request for “better ideas” does not tell Claude what better means. A request for options with different emphases gives you a comparison that you can inspect.

Review against stated criteria

Criteria are checks that you use to judge an answer. Write them before you ask Claude to review its work. They can include facts from the task, such as the time limit, and qualities that you want, such as plain language or a privacy reminder in each activity.

For the class plan, the criteria might be: it lasts 60 minutes, works for both iPhone and Android users, uses no paid apps, includes a privacy reminder, and gives beginners enough practice time. A criterion should be specific enough that you can explain why a plan passes or needs work. For example:

Review the [chosen version] against these criteria: 60 minutes total; suitable for beginners using iPhone or Android phones; no paid apps or extra equipment; at least 20 minutes of practice; and a privacy reminder in every activity. Use a table with the criterion, pass or needs revision, evidence from the plan, and any question that I still need to check. Do not claim that the plan passes when information is missing.

Claude can help perform the review, but you remain responsible for the decision. Check its timing, calculations, facts, and quotations. For work that affects safety, money, health, legal duties, employment, or another person’s rights, use suitable qualified human review.



Watch out: Asking Claude to review its own work against your criteria is still Claude judging Claude. It can mark a weak answer as passing. Spot-check at least one criterion yourself instead of accepting every “pass” at face value.

Revise one part without losing the rest

A follow-up can change one section without starting the task again. Name the part that needs work, state what must remain unchanged, and ask to see only the changed section when that makes comparison easier.

Suppose the review shows that the plan has only 10 minutes of practice. You can ask Claude to adjust the timing while keeping the audience, privacy boundary, and the activities that already work:

```
Revise only the timing and activities needed to provide at least 20 minutes of hands-on practice. Keep the total at 60 minutes. Keep the audience, standard-phone-features rule, privacy reminders, and the existing opening activity. Show only the changed rows, then give the revised total time.
```

Read the changed rows and compare them with the earlier plan. Claude may still alter more than you asked it to change. If that happens, point to the changed item and ask it to restore the earlier version. For longer work, copy approved text into your own document and keep a dated version before you request a major rewrite.



Good practice: Use a new chat when you change to a different subject. Use the same chat when the next request depends on its earlier messages. If you must begin a new chat, paste the short task summary, the criteria, and the current version that you want revised.



Learn more online: *Prompt engineering overview:* <https://docs.claude.com/en/docs/build-with-claude/prompt-engineering/overview>.

Try it now

Write one prompt in four parts, then improve it through conversation. This activity should take 10 to 20 minutes. Choose a personal planning, learning, or writing task. Do not upload a file or enter private information:

1. Start a new chat. Choose a task with a visible result, such as a study plan, a packing checklist, a comparison table, or a draft agenda.
2. Write one sentence that states the task and desired outcome. Add two or three facts that provide relevant context, one or two constraints, and a boundary if needed.
3. Ask for a specific form of answer, such as a table with named columns or a numbered checklist. Send the prompt.
4. Ask Claude to list missing information that could change the result. Answer the useful questions.
5. Ask for two or three alternatives that differ in one named way. Choose one and ask Claude to review it against two to four of your own criteria.
6. Request one focused change. Ask to see only the changed section, then compare it with the earlier version before you keep it.



What to ask next: Summarize the decisions in this conversation. List the information I supplied, the assumptions that remain, the criteria that still need a human check, and the next action I should take.

Check the result

- ☐ Did your prompt name both a task and a result you can review, using context that changes the answer without adding private or unrelated details?
- ☐ Did you state a limit or boundary that protects your time, money, privacy, or work?
- ☐ Did you ask for missing information instead of letting Claude fill every gap with a guess, and did the alternatives differ in a named way you could compare?

If an answer is no, revise that part of the prompt and compare the next response. You are learning how to give Claude useful instructions, not trying to write a perfect message on the first attempt.

Chapter 1.6 uses this same habit of staying in the conversation and turns it toward learning: asking for an explanation at the right level and checking your own understanding instead of only reading Claude's answers.

Chapter 1.6 Learn Something New or Tutor a Child

Chapter 1.5 showed how to keep a conversation going until a task's result is right. That same habit works for learning, whether you're the one learning or you're helping a child with schoolwork. Claude can act as a patient tutor for almost any subject: a language, a craft, a school topic, or a skill you need for work.

In this chapter, you will ask for an explanation pitched to your level, break a subject into a short learning path, request analogies and practice questions, let Claude question you instead of only answering, and check your own understanding by explaining an idea back in your own words. You will then use those same habits to help a child with homework without doing it for them.

Ask for an explanation at the right level

A good teacher asks what you already know before explaining something new. Claude cannot see your background unless you state it. The same question can need a different answer for a beginner and for someone who already knows the basics.

To help Claude to help you learn something new:

1. Say what you can already do.
2. Name the specific part that confuses you.
3. Set a length so the answer stays focused.

Imagine you are learning Spanish before a trip and the difference between the verbs *ser* and *estar* keeps tripping you up:

I know how to introduce myself and order food in Spanish, but present tense verbs still confuse me. Explain when to use *ser* and when to use *estar*, using words I already know. Keep the explanation under 200 words and include two example sentences.

Ser vs estar in Spanish ▾ Share

1

I know how to introduce myself and order food in Spanish, but present tense verbs still confuse me. Explain when to use ser and when to use estar, using words I already know. Keep the explanation under 200 words and include two example sentences.

2

3

Here's the core trick: **ser** is for things that stay the same, **estar** is for things that can change.

Use SER for:

- Who someone is (name, job, nationality) — like introducing yourself
- What something is made of or its basic nature

Use ESTAR for:

- Where something is located

↓

Figure 1.6A: Say what you can already do, name the specific part that confuses you, and set a length so the answer stays focused

If the reply still is not clear, say which part did not make sense. Ask for the same idea explained a different way, such as with a shorter answer, a different example, or fewer new words.

Break a difficult subject into a learning path

A single explanation can cover one idea, but a subject usually has several parts that build on each other. Ask Claude to lay out a short sequence of topics in the order you should learn them, instead of trying to absorb everything at once:

I want to hold a simple conversation in Spanish on a trip in three weeks. Break this into a short learning path. List the grammar and vocabulary topics in the order I should learn them, and mark which topics depend on an earlier one.

Greetings & Introductions

Basic pleasantries (hola, buenos días) and self-intro phrases like 'me llamo' and 'soy de'. You already have a head start here — this is your foundation.

1

2

3

4

5

6

7

8

View all steps

Next

That gives you a clear runway before your trip — roughly one topic every 2-3 days over three weeks. The two verb-heavy steps (present tense conjugation and irregular verbs) will take the most practice, so it's worth drilling those with flashcards or a conjugation app rather than just reading through them once.

Figure 1.6B: A step-by-step learning path. Select Next or Back, or the numbered buttons, to navigate the steps

Read the path before you follow it. Ask Claude to explain why one topic comes before another if the order is not obvious. You can also ask it to shorten the path if you have less time, or to mark which topics you can skip for your specific goal.

Request examples, analogies, and practice questions

An explanation is easier to remember when it links to something you already understand, and easier to use when you get to practice it. Ask Claude to compare the new idea to something familiar, then ask for a short exercise that uses it:

Compare the Spanish subjunctive mood to something in English grammar that I probably use without noticing. Give one short example in each language. Then write two practice sentences with a blank where I need to choose the correct verb form.

Check whether the comparison actually fits. A comparison can make an idea memorable while still leaving out an important difference:

Where does that comparison break down? Give one case where English and Spanish handle this differently.

Let Claude question you, rather than reveal the answer

Reading an explanation is not the same as learning it. Answering a question about it is what helps the idea stick. Ask Claude to write questions without the answers, answer them yourself, and then ask it to check your work:

Write five short quiz questions about *ser* and *estar*. Do not include the answers. Wait for me to answer each question before you tell me if I am right, and explain any mistake.

Write five short quiz questions about *ser* and *estar*. Do not include the answers. Wait for me to answer each question before you tell me if I am right, and explain any mistake.

Here's question 1 — I'll wait for your answer before moving to the next.

1. Fill in the blank: Yo _____ profesor. (*ser/estar* — choose the correct form)

Figure 1.6C: Asking for a quiz on Spanish grammar

When you answer, Claude will correct you if needed, and then continue with the quiz:

ser

Correct! "Yo soy profesor" uses **ser** because being a teacher is a lasting fact about identity/occupation, not a temporary state.

2. Fill in the blank: El café _____ frío. (*ser/estar* — choose the correct form and conjugate it)

Figure 1.6D: Claude will correct you and then continue with the quiz

Claude will sometimes reveal an answer inside a later question or explanation if you do not ask it to hold back. If that happens, ask for a fresh set of questions and try again.

Explain it back to check your understanding

Once you can answer quiz questions, try the harder test of explaining the idea in your own words. Write your explanation, then ask Claude to check it for gaps or mistakes instead of grading you against its own version first:

```
I will explain when to use ser versus estar in my own words. Read what I write and tell me what is correct, what is missing, and what is wrong: [type your explanation here]
```



Good practice: A day or two after Claude checks your explanation, try explaining the same idea again without looking at the chat. If you cannot restate it without checking the chat, that topic needs another round of practice.

Verify facts and keep your own thinking in charge

Claude can state a grammar rule, a translation, or a fact with confidence even when it is wrong or does not fit your specific region or dialect. Before you rely on a phrase in person, such as ordering food or asking for help, check it with a native speaker, a textbook, or a course you trust.

Use Claude to explain, quiz, and check your work, not to think through the problem for you. Try an exercise on your own first, then ask Claude to review your attempt. This keeps the practice in your hands and gives Claude a specific, useful job: checking your reasoning instead of replacing it.



Watch out: A confident-sounding answer is not automatically an accurate one. Treat a surprising fact, an unfamiliar rule, or anything you plan to say to another person as worth a second source.

Use Claude ethically for schoolwork and essays

Nowhere does the habit from this chapter – explaining, quizzing, and checking rather than answering for you – matter more than a graded assignment. Using Claude the wrong way on an essay can cost you or a child you love an academic integrity case, not just earn a weak grade.

Read the AI policy for the specific course and assignment, as well as your school's general policy. Course-level rules increasingly differ from each other even within the same department, and a general school policy may not settle what your instructor actually expects. If nothing is written down, ask before you start, and keep a record that you asked.



Good practice: Write your own first draft of an academic essay assignment before asking Claude to react to it. Getting feedback on your own attempt teaches you more than asking Claude to produce the attempt for you, and it keeps the finished essay defensibly yours.

There's a line between support and substitution. Asking Claude to brainstorm angles, build an outline, explain a source, check grammar, or react to a draft you already wrote is learning support. Asking it to write paragraphs, or the whole essay, that you then submit as your own is not, even at schools that otherwise welcome AI use, because it misrepresents someone else's writing as yours.

If your instructor asks you to disclose AI use, be specific rather than vague. "I used Claude to build an initial outline and check grammar in my final draft" tells your instructor something they can evaluate. "I used AI" does not and reads as evasive even when it's honest.



Real-world example: Before writing a response to an assigned reading, a student uses Claude to quiz them on the reading's argument until they can restate it correctly, then writes the analysis entirely themselves and adds a short note describing that use, matching their instructor's disclosure requirement.



Watch out: Submitting text Claude wrote as entirely your own work counts as academic misconduct at most schools, whether or not a detector flags it, and detection tools are unreliable. This isn't hypothetical: in 2026, a publisher canceled a debut novelist's roughly \$2 million book deal after the author's own team said they could no longer verify how the manuscript was actually written, regardless of whether AI was involved. "Will it get caught?" is the wrong question. The one that matters is whether you can honestly defend the work as genuinely yours.



Learn more online: Starting with models released on or after August 2, 2026, Anthropic began embedding machine-readable watermarks into text generated by Claude. The text watermark works by subtly biasing Claude's word choices in detectable patterns and travels with the text even when copied and pasted elsewhere. *How Claude marks AI-generated content:* <https://support.claude.com/en/articles/16266773-how-claude-marks-ai-generated-content>.

Helping a child with their homework

Homework help is one of the most common reasons a parent opens Claude in the evening, and one of the easiest places to slip from teaching into doing. After a long day, it is tempting to paste your child's worksheet question into the chat, copy back the answer, and move on. That does not help your child learn, and it leaves them stuck the next time a similar question shows up on a test with no Claude beside them.

Treat Claude the same way this chapter treats it everywhere else: as a tutor that questions and explains, not a source of finished answers you relay. You should stay in the room, read what Claude asks and says, and decide what your child sees.

Children copy how the adults around them use a tool as much as what those adults tell them. Talk through what you are doing as you use Claude with, or in front of, your child: "I'm asking it to check my explanation before I explain this to you," or "I'm going to ask it to question you instead of giving the answer." That running commentary teaches the habit you want your child to build later, more than a rule about screen time will.

Math and science: ask for a method, not a result

Ask Claude to work alongside your child rather than solve for them:

My child is working on this algebra problem: [paste the problem]. Do not give the final answer. Ask one guiding question at a time, and only offer a hint if they are still stuck after two attempts.

Check the working, not just the final number. A child who reaches the right answer from a hint they did not understand will struggle with the next problem that uses the same idea.

English essays: feedback on their draft, not a draft from scratch

The child should write their essay first, and then ask Claude to react to what they wrote:

My child (age 12) wrote this paragraph for a book report: [paste the paragraph]. Do not rewrite it. Ask two questions that would help them add more detail, and point out one thing that already works well.

History and art: use the parts a worksheet can't mark

History and art homework often plays to what Claude is genuinely good at: quizzing dates and causes back to your child, suggesting a few directions for a project, or explaining a technique they noticed in a museum photo. Ask for questions and options here too, rather than a finished paragraph or a ready-made “five causes of X” list your child can paste in unchanged.

Check the assignment's own rules first

Before you open Claude for a specific assignment, check what the teacher and school actually allow. Some homework is meant to be closed-book practice; some explicitly welcomes AI tools. When you are not sure, have your child ask their teacher rather than guessing.



Watch out: The hardest version of this habit to catch is reaching for Claude because a deadline is close, not because your child is stuck on the material. If you notice you are using it to finish something quickly rather than to help your child understand it, that is the moment to slow down, not speed up.

Try it now

Use Claude to start learning one topic or skill. This activity should take 15 to 20 minutes. Choose a subject you want to learn for work, school, or personal interest:

1. Start a new chat. State your goal in one sentence and describe what you already know about the topic.
2. Ask for a short learning path: the topics to cover, in order, and which ones build on each other.
3. Pick the first topic. Ask for an explanation at your level, an analogy, and one or two practice questions.
4. Ask for three to five quiz questions without answers. Answer each one, then ask Claude to check your answers and explain any mistakes.
5. Explain the topic back in your own words and ask Claude to point out what is missing or wrong.



What to ask next: Based on what I got right and wrong, tell me which topic I should study next and why.

Check the result

- ☐ Did you tell Claude what you already know before asking for an explanation?
- ☐ Does your learning path show which topics come first and why?
- ☐ Did you answer the quiz questions yourself before Claude checked them?
- ☐ When you explained the topic back in your own words, could Claude confirm it was accurate and complete?

If an answer is no, return to the chat and ask Claude to try that part again. Whether you're learning something new yourself or helping a child do the same, the habit is the same: ask, question, and check, rather than just answer.

Chapter 1.7 turns from learning a subject to planning everyday decisions: explaining your circumstances, comparing options, and testing assumptions before you choose.

Chapter 1.7 Plan Everyday Life and Compare Important Choices

You will use Claude to think through a real decision instead of asking one question and stopping there. This works for choosing a product, planning a trip, organizing a home project, preparing meals from what is already in your kitchen, planning a new exercise routine, or preparing questions before you see a doctor, a lawyer, an accountant, or a financial adviser.

Before you begin, pick one decision you are actually facing. A real decision, even a small one, teaches this skill better than a made-up example. By the end of this chapter, you will have a short, organized comparison you can use to decide, or to take into a conversation with a person who can advise you.

Explain your circumstances, goals, and constraints

Start by describing your situation in your own words. Include what you want to achieve, what limits you, such as budget, time, or location, and anything that matters to you personally.

For example:

```
I'm planning a five-day trip for two adults and one child. Our budget is US$1,500 excluding flights. We want somewhere warm in October, and my child gets bored on long car journeys.
```

The more relevant detail you give, the more useful the reply. Leave out anything private that you would not want stored, such as your exact address or account numbers:

I'm planning a five-day trip for two adults and one child. Our budget is US\$1,500 excluding flights. We want somewhere warm in October, and my child gets bored on long car journeys.

Worked for 23s >

Best fit: Costa Adeje, Tenerife

Assuming your US\$1,500 budget covers accommodation, meals, local transportation, and activities for five days, I would choose Costa Adeje in southern Tenerife.

It suits your requirements because:

- October is normally warm enough for beaches and pools, with typical daytime temperatures around 21°C to 26°C (70°F to 79°F) and relatively little rain.  +1



Figure 1.7A: A clear starting message names the goal, the budget, and the one constraint that matters most: a child who dislikes long car journeys

Ask Claude to identify missing information

Before you ask for a recommendation, ask what else would help. A short prompt does this well:

Before you suggest anything, tell me what other information would help you give a better answer.

Claude might ask about your travel dates, how you feel about flying, or whether you have visited similar places before. Answer the questions that matter to you and skip the rest.



Good practice: Treat these follow-up questions the way you would treat questions from a helpful travel agent or shop assistant. You do not owe a complete answer to every one, but each answer you give narrows the search.

Generate options before asking for a recommendation

Ask for several options before you ask which one is best. This keeps you from anchoring on the first idea you hear. For example:

Give me three different options that fit what I've described. Don't recommend one yet. List the choices only.

Reviewing several options side by side shows you the real range of choices before you narrow down to one.

Compare options using explicit criteria

Once you have options, name the criteria that matter to you and ask Claude to compare the options against them. Criteria might include cost, time, effort, risk, or how well an option fits a constraint you named earlier:

Compare these three options against cost, travel time, and how well they suit a six-year-old. Use a table.

A table makes trade-offs easier to see than a paragraph of prose. Ask for one whenever you are comparing more than two or three options. The table might look like this:

Goals	Constraints	Evidence	Trade-offs	Unanswered questions
Orlando, Florida Warm pools and theme parks	US\$1,500 for 5 days, no long drives	Warm October weather, short airport-to-hotel transfer	Higher attraction costs, busy resort areas	Which airport? Do you want a theme-park day?
San Diego, California Beach and animals	US\$1,500 for 5 days, no long drives	Mild, sunny October, zoo and beach close together	Usually less hot, hotel prices vary	Is mild warmth enough? Do you need a pool?
Cancun, Mexico Beach resort break	US\$1,500 for 5 days, no long drives	Warm sea and resort activities, airport transfers available	Weather can be rainy, passport and resort costs	Are passports ready? All-inclusive or self-catering?

Table 1.7: A choice matrix separates what you want from what you know, what you would give up, and what you still need to find out

Test assumptions and consider disadvantages

Every option has a downside. Ask directly for the disadvantages and question any assumption you are not sure about:

What could go wrong with option 2? What am I assuming that might not be true?

This step matters most when an option sounds appealing. A confident-sounding answer is not the same as a well-supported one. Ask Claude to explain its reasoning, and check anything that surprises you against another source before you rely on it.



Watch out: Claude can describe disadvantages you did not ask about only if you prompt for them. Left alone, it will often lean toward the option that best matches your stated preferences, so ask it to argue against your likely choice at least once.

Separate low-stakes guidance from decisions requiring qualified advice

Some decisions carry little risk if you get them wrong, such as which restaurant to try or how to pack for a trip. Other decisions, including anything about your health, your legal rights, your taxes, or your safety, deserve a qualified professional's review before you act.

Claude can help you prepare for that conversation. It can help you list your symptoms clearly before a doctor's appointment, draft questions for a solicitor or lawyer, or organize your finances before you speak with an accountant. It should not replace the appointment itself.

Ask Claude to help you prepare, then confirm with a qualified person.



Figure 1.7B: As the stakes rise, Claude's role shifts from adviser to a way of preparing for advice from a qualified person

Try it now

Choose a real decision you are currently facing, even a small one. Describe your goals and constraints, ask what information is missing, request several options, compare them against your own criteria in a table, and ask what could go wrong with your likely choice.



What to ask next: Turn this comparison into a short checklist I can bring to a qualified person. List the questions I should ask them and the facts I should confirm myself first.

Check the result

- ☐ Did the comparison use criteria that actually matter to you, not just the ones Claude chose?
- ☐ Did you ask for disadvantages, or only see the advantages of each option?
- ☐ If this decision involves your health, legal rights, money, or safety, do you have a plan to confirm it with a qualified person?

The comparison you just built rests on Claude's own reasoning.

Chapter 1.8 adds a further check. You will learn when a task needs current information from the web, how to make sure facts such as prices and opening dates are up to date, and how to inspect the sources Claude uses.

Chapter 1.8 Search and Verify Current Information

You will ask Claude a question that needs current information, and then check the sources it used before you trust the answer. Do this for anything that changes often, such as prices, news, opening hours, availability, or a person's current job title, because Claude's stored knowledge can be out of date.

Before you begin, think of a question you would normally look up online, such as whether a store is open today or what a recent news story actually said. By the end of this chapter, you will know when Claude searches the web on its own, how to open and read its sources, and how to check that a source really supports what Claude told you.

Know when web search is necessary

Claude can answer many questions using what it learned during training. That training has a cutoff date, so it cannot know about anything that happened after that point on its own. For anything current, Claude needs to search the web while answering you.

When **Web search** is enabled, Claude can decide to search when your question would benefit from current information. Words such as “today,” “latest,” “current,” or “price” are strong clues that current information may be needed. You can also ask for a web search explicitly if you are not sure it happened:

```
Search the web for this and show me your sources: What are the current opening hours for [a specific business]?
```

For example, I might search for a local burger joint in my hometown of Brighton:

```
Search the web for this and show me your sources: What are the current opening hours for Smashburger in Brighton and what special offer do they have today?
```

Confirm that web search is enabled by clicking the + in the chat box. The **Web search** menu should be checked to show that it is enabled:

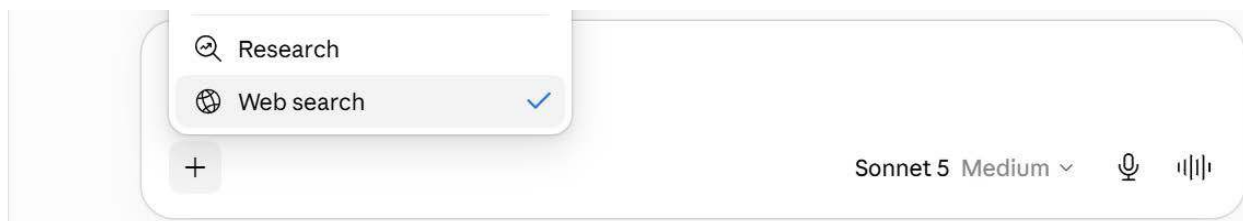


Figure 1.8A: Web search is enabled for this chat



Good practice: If you suspect an answer could be outdated, such as a price, a schedule, or a recent event, ask Claude to search rather than assuming it already has. Check all sources before trusting the supported answer.



Learn more online: *Enable and use web search:* <https://support.claude.com/en/articles/10684626-enable-and-use-web-search>.

Ask for useful and varied sources

A single source can be wrong, outdated, or one-sided. Ask Claude for more than one source, and ask for sources of a particular kind when it matters, such as an official website, a news report, or a government page:

```
Give me three sources for this, including at least one official or primary source, and tell me how they differ.
```

You should now see a response with visible citations. If Claude gives you only one source, or sources that all repeat the same original article, ask for a wider search before you rely on the answer.

Open and inspect citations

When Claude uses search, it usually adds links to the pages it used:

1. Read the response as usual.
2. Click a link to see its source.
3. Open at least one source yourself for any claim you plan to rely on.

Searched the web

Here's what I found for the Smashburger branch at 53 North Street, Brighton (BN1 1RH).

Opening hours today (Wednesday): 11am - 10pm

Today's offer (Wednesday): Kids Eat Free — a kid's beef, chicken, veggie, or vegan burger (or tenders) plus fries and a drink, free with the purchase of an adult meal. It's dine-in only, ordered at the till (not available via the self-service kiosk).

Sources:

1. **Smashburger UK official site — Locations page** (primary/official):
smashburger.co.uk/locations — lists the Brighton branch's hours as Sun–Thurs 11am–10pm, Fri & Sat 11am–midnight, confirming today's (Wednesday) hours as 11am–10pm.
2. **Smashburger UK official site — Kids Eat Free page** (primary/official):
smashburger.co.uk/kids-eat-free — confirms the "Kids Eat Free" promotion runs every

Figure 1.8B: The response starts with “Searched the web” and includes links to citation sources

Verify dates, claims, and quotations

For every link, open the source and check three things: the publish or update date, whether the page truly supports the specific claim, and whether any quoted words match the source exactly.



Watch out: A confident answer with a linked source is not automatically accurate. Claude can summarize a source incorrectly or attach the wrong page to a claim. Treat every citation as a starting point for your own check, not as proof on its own.

For anything you plan to repeat, publish, or act on, such as a statistic in a report or a quotation in an email, open the original source and read the relevant passage yourself.

Try it now

Ask Claude a question about something current, such as a recent price, a news event, or an organization's current contact details. Request three sources, click through to at least one source, and check whether it actually supports the claim Claude made.



What to ask next: Show me how you decided this needed a web search instead of answering from memory. Then tell me one kind of question where you would not need to search.

Check the result

- ☐ Did Claude search the web, or did it answer from memory alone?
- ☐ Do the sources come from more than one website?
- ☐ Does the source you opened actually say what Claude claimed, including any dates or numbers?

A single search answers a single question well. *Chapter 1.9* covers research, a built-in feature for questions that need several sources and more time to work through properly.

Chapter 1.9 Conduct Research with Cited Sources

You will ask Claude to run a longer research task instead of a single search. This built-in feature is called **research mode**. Rather than answering right away, it plans its own steps, reads many sources over several minutes, and returns a structured report with citations.

To enable research mode, you must:

1. In the chat box, select the + menu and select **Research**.
2. Note the chat box now shows an icon to indicate research mode is enabled. You can disable research mode by selecting its x button to remove it from the chat box:

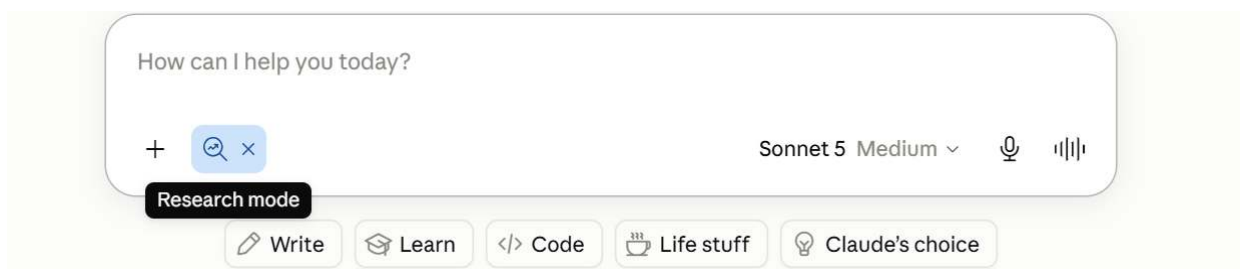


Figure 1.9A: Research mode is enabled

Research mode suits questions that a single search cannot answer well, such as comparing several options, understanding a topic with conflicting evidence, or preparing a briefing that needs multiple sources.



Watch out: Research currently requires a paid Claude plan such as Pro, Max, Team, or Enterprise. Because a Research session performs multiple searches and processes many sources, it can consume your usage allowance faster than an ordinary conversation. Check your plan's current limit before you start, since it can change.



Learn more online: Use research on Claude: <https://support.claude.com/en/articles/11088861-use-research-on-claude>.

Before you begin, have a specific question ready. By the end of this chapter, you will have run one research task and reviewed its report against its own sources.

Define a research question with scope and request a plan

A vague question produces a vague plan. Narrow your question before you start and decide how wide the research should range.

State the topic, the time period that matters, and any boundary you want respected. For example, rather than “Tell me about electric cars,” try:

Research the main barriers to electric car adoption in Europe over the past two years, focusing on charging infrastructure and price.

A clear scope also tells Claude what to leave out, which keeps the final report focused on what you actually need.

Activate **Research** before you type or send your prompt. A research task can take several minutes to finish, so be patient:

Research the main barriers to electric car adoption in Europe over the past two years, focusing on charging infrastructure and price. Before you start, show me your research plan so I can check it.

Here's my proposed research plan for this topic:

Scope: Barriers to EV adoption in Europe, past ~2 years (mid-2024 to present), focused on two main drivers:

1. Charging infrastructure

- Public charger density/coverage across EU countries (and UK), including rural vs. urban gaps

Figure 1.9B: Research proposes a plan before it searches. Reviewing it at this early stage is faster than fixing a finished report

Read, copy, download, or publish the finished report publicly:

Electric car adoption barriers in Europe ▾

EV adoption barriers in Europe >

Research complete • 273 sources • 7m 12s

Your report on barriers to electric car adoption in Europe (mid-2024 to August 2026) is ready. It examines how charging infrastructure and price remain the dominant barriers, with analysis of the AFIR rollout, geographic concentration of chargers (Netherlands, Germany, and France hold ~61%), reliability and payment fragmentation issues, and the €1,800 drop in average BEV prices in 2025.

Copy ▾ Publish ✕

Download as Markdown
Download as PDF

Barriers to Electric Car Adoption in Europe, mid-2024 to August 2026: Charging Infrastructure and Price

TL;DR

- Charging infrastructure is improving faster than sales but remains geographically lopsided and operationally unreliable: Europe passed roughly 1.15 million public charging points by end-2025,

Figure 1.9C: Read, copy, download the finished report as a Markdown or PDF file, or publish.



Watch out: Publishing this artifact will make it accessible to anyone on the internet and potentially visible in search engine results. Your chat will remain private.

There are several techniques to get better research results.

Set source and date requirements

Tell Claude what kind of sources you want and how current they need to be. This is especially useful for topics where outdated or low-quality sources would mislead you.

Use sources published in the last 18 months where possible. Prefer official statistics, peer-reviewed research, and established news organizations over blogs or forum posts.

If your topic depends on very recent developments, say so directly, since research otherwise decides source recency on its own.

Turn findings into a useful conclusion

Ask for a report structured around your original question rather than a plain list of facts. A useful structure includes a short summary, the main findings grouped by theme, any contradictions between sources, and what remains unclear:

Summarize your findings in four sections: key findings, points where sources disagree, remaining gaps, and a one-paragraph conclusion.

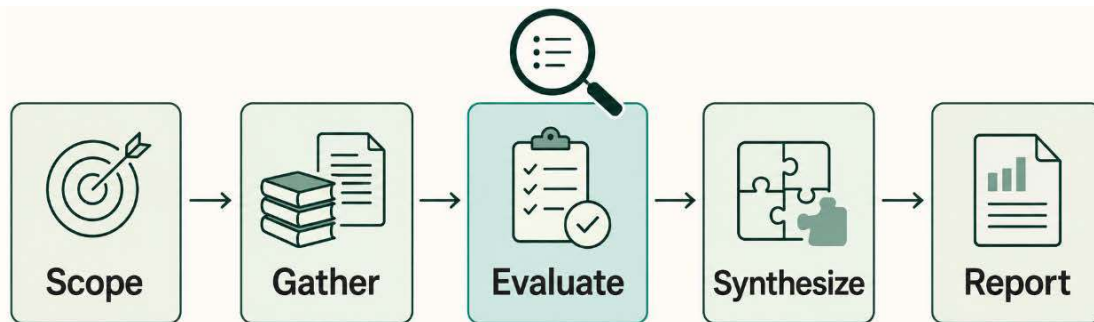


Figure 1.9D: Research moves through these five stages on its own. Your scope at the start and your review at the end are the two points where you stay in control.



Watch out: A longer, more thorough-looking report is not automatically more accurate than a short one. Research can still misread a source or overstate a weak finding. Treat its report as a strong first draft that still needs your review.

Read the report the way you learned in *Chapter 1.8*: open a few of the cited sources yourself, check that they say what the report claims, and note the publication dates.



Good practice: Save a finished report you plan to reuse. You will build on this skill in *Chapter 1.11*, where projects keep research material alongside related conversations.

Try it now

Choose a question that needs more than one source to answer well. Define its scope, ask Claude to show its plan before starting, set a date and source requirement, and request a report with key findings, disagreements, and gaps.



What to ask next: Turn this report into a one-page briefing for someone who has not read the full research. Keep the key findings and disagreements, and flag anything that still needs a human decision.

Check the result

- ☐ Did the final report match the scope you set, or did it wander into unrelated territory?
- ☐ Did you review the plan before the research ran, or only the finished report?
- ☐ Did you open at least one cited source yourself, and did it support the claim attached to it?

Research works from sources Claude finds on its own. *Chapter 1.10* turns to sources you already have: your own documents, photos, and spreadsheets.

Chapter 1.10 Add Documents and Images to a Chat

You will upload a file, a photo, or a spreadsheet and ask Claude to work with it directly, instead of typing everything out yourself. This includes documents such as contracts and reports, photos of receipts or whiteboards, and structured data such as a spreadsheet or CSV file.

Before you begin, have one file ready to try, something ordinary, such as a bill, a short report, or a spreadsheet with a few rows of data. By the end of this chapter, you will know how to upload a file, ask a clear question about it, and get a useful, checkable result back.



Watch out: The free plan allows a limited number of file and image uploads per day. Paid plans raise these limits. The exact numbers change from time to time, so check the book's companion page or your account settings for the current limits.

Upload files and explain the assignment

Click + in the chat box, select **Add files or photos**, and then choose a file from your device or a connected cloud storage account. Claude accepts common document formats such as PDF and Word files, image formats such as JPEG and PNG, and spreadsheet formats such as Excel and CSV. Excel XLSX files require code execution and file creation to be enabled for your account.



Sample files: If you do not have your own files then you can download a sample file from the following link: <https://github.com/markjprice/claude-vb/tree/main/Files/Book1/1-10-Two-Page-Rental-Agreement.pdf>.

Uploading a file is not enough on its own. Explain what you want done with it, just as you would if you handed a document to a colleague. State the purpose, the audience if there is one, and the form you want the answer in:

This is a two-page rental agreement. Summarize the key obligations for the tenant in plain language, and flag anything unusual compared to a standard lease.



Figure 1.10A: Naming the file's purpose produces a far more usable result than uploading a file with no instructions

Here is the response from Claude:

Here's a plain-language summary of the tenant's key obligations, plus what stands out as unusual.

Tenant's core obligations

Rent & money

- \$1,850/month, due the 1st; \$75 late fee if not paid by 5 p.m. on the 5th; \$35 fee for returned payments
- \$1,850 security deposit before move-in



Figure 1.10B: Claude's response

Watch out: Do not upload passwords, private keys, medical records, or anyone else's personal information unless you have the right to share it. Check your organization's rules before uploading work documents, a topic covered fully in *Chapter 2.1*.

Learn more online: *Upload files to Claude:* <https://support.claude.com/en/articles/8241126-upload-files-to-claude>.

Summarize and extract information

Ask for a summary written for a specific reader rather than a general one. A summary for yourself can stay informal. A summary for a manager or a client should be shorter and focus on decisions and numbers.

To pull specific details out of a longer document, ask for them by name and request a structured format:

Extract every date, dollar amount, and named party from this contract, and put them in a table with columns for Item, Value, and Page.

Here's the full extraction:

Item	Value	Page
Landlord	Example Property Group LLC	1
Tenant	Jordan Lee	1
Premises address	Apartment 4B, 123 Example Street, Harbor City, CA	1

Figure 1.10C: Naming the exact output you want, here a table, produces a far more usable result

A table is easier to scan and easier to check against the original document than a paragraph.

Compare multiple documents

Upload two or more files in the same conversation and ask Claude to compare them directly. This works well for comparing quotes from different suppliers, two versions of the same document, or a set of resumes against a job description.

I've attached three supplier quotes. Compare them on price, delivery time, and warranty terms, and show the comparison as a table.

Name the specific points of comparison you care about. Without direction, Claude may compare details that matter less to you and skip the ones that matter most.

Analyze images and structured data

Claude can look at a photo and describe what it sees, read printed or handwritten text, and answer questions about a chart or diagram in an image:

Here's a photo of a whiteboard from our meeting. Transcribe the text and organize it into a list of decisions and action items.

Handwriting, glare, and blurry photos reduce accuracy, so review the transcription against the original image, especially for numbers and names.

For spreadsheets and other structured data, Claude can clean messy columns, calculate totals, and build a chart from the numbers you provide:

Here's a spreadsheet of monthly expenses. Clean up any inconsistent category names, then show total spending by category as a bar chart.

You should now see a chart or table built directly from your uploaded data. *Chapter 2.5* covers spreadsheet analysis in more depth, including checking formulas and calculations.



Good practice: Keep the original file open next to Claude's result. Compare the two side by side so you can catch a misread number, a dropped row, or a merged cell before you use the extracted version.

Take a screenshot

You can capture part of or your entire screen and attach it directly to a chat without first saving an image file. This is useful when you want to ask Claude about an error message, webpage, document, dialog box, or anything else currently visible on your screen.

In the chat box, select **+** and then **Take a screenshot**. When the **Choose what to share with claude.ai** dialog box appears, select the **Microsoft Edge Tab**, **Window**, or **Entire Screen** that you want Claude to see. The captured image is automatically attached to your message. You can then add a question such as:

What does this error message mean, and how can I fix it?

Claude Desktop needs permission to record your screen for these features to work.



Good practice: Capture only the part of the screen Claude needs. A smaller, focused screenshot makes it easier for Claude to identify the relevant text or interface elements and reduces the chance of accidentally sharing private information.

Try it now

Upload one file, a document, a photo, or a spreadsheet, and give Claude a specific instruction that names the purpose and the format you want back, such as a summary, a table, or a chart.



What to ask next: Check your own extraction against the original file one more time. List anything you are less than fully confident about, such as unclear handwriting, a merged cell, or an ambiguous date format.

Check the result

- ☐ Did you tell Claude what the file was and what you wanted done with it?
- ☐ If Claude extracted numbers, dates, or names, do they match the original file?
- ☐ Did you avoid uploading anything private or confidential that you should not share?

Every skill so far has lived inside a single conversation. *Chapter 1.11* shows how to save your preferences and your recurring work so you stop repeating yourself in every new chat.

Chapter 1.11 Organize with Instructions, Memory, and Projects

You will set up Claude so it remembers what you want it to remember, forgets what you would rather keep separate, and keeps a recurring piece of work organized in one place. This removes the need to repeat your background and preferences in every new conversation.

Before you begin, think of one task you return to often, such as planning meals, preparing weekly reports, or studying a subject. By the end of this chapter, you will have personal instructions set, a clear idea of what Claude remembers about you, and one project ready to hold that recurring task. You will build on this project in *Chapter 1.12*.



Watch out: Projects are available on the Free plan as well as paid plans, although Free accounts currently have a limit on the number of projects they can create. Cross-conversation Memory requires a supported paid plan, so some readers may not see all of the Memory controls shown in this chapter.

Set durable personal instructions

Some facts about you rarely change: your job, your preferred tone, or a format you always want. Rather than repeating these in every conversation, save them once as **Instructions for Claude**. Claude applies them automatically to every new chat:

1. Select your account menu, and then **Personalization**, or you can select **Settings** and then **Personalization**.
2. Scroll down the settings to find **Instructions for Claude**.
3. Describe your role, your preferences, and anything Claude should always avoid, and then click **Save changes**:

I'm a secondary school teacher. Keep answers concise and avoid jargon. When you give me a lesson idea, include one way to check whether students understood it.

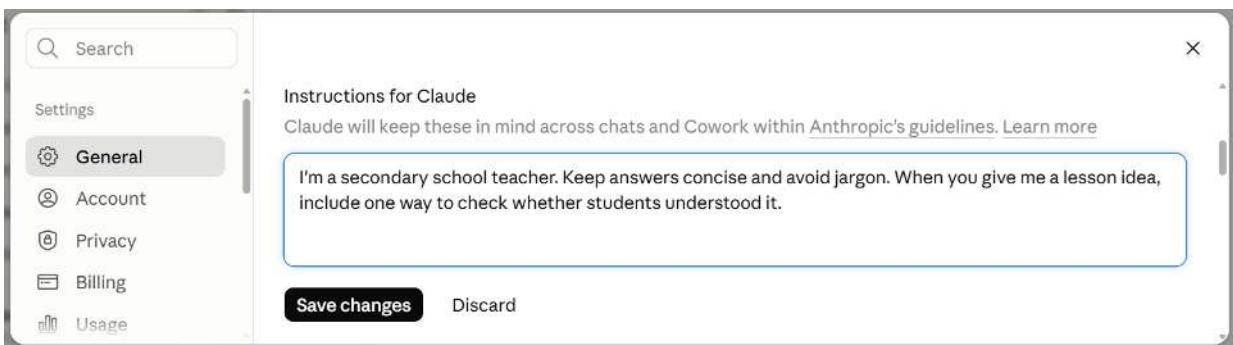


Figure 1.11A: General settings and instructions for Claude

You should now see your instructions saved, and a new conversation should reflect them without you mentioning your role again.



Watch out: Instructions for Claude and memory apply automatically, even when you forget they are switched on. A reply that seems unusually short, informal, or oddly specific to your situation may be following a saved instruction or memory rather than your prompt. Check **Settings | General | Instructions for Claude** if a response looks unexpectedly shaped by something you didn't just say.

Decide what Claude should remember

Instructions for Claude are rules you set once. **Memory** is different: it is made of specific facts Claude picks up while you talk, plus what it can recall from your earlier conversations. Together they let Claude respond as someone who already knows your situation, instead of starting from zero each time.

You can add a memory directly by telling Claude to remember something:

Remember that I'm training for a 10K race in October and I prefer running in the early morning.

Memory works best when it stays accurate. Check it occasionally and remove anything outdated or wrong.

To edit Claude's memories of you:

1. Select your account menu, and then **Settings**.
2. In the **Customize** section, select **Memory**, and review the memory summary about you in **Areas**, like the **Weekend Workshops** memory it recorded about my earlier conversation with it:

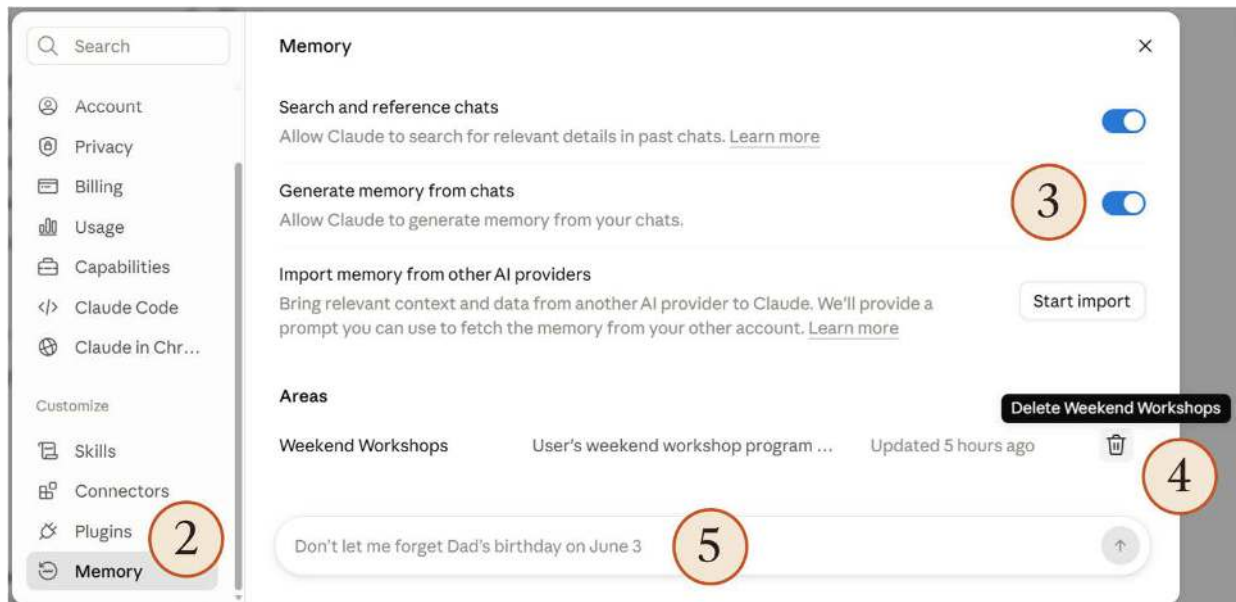


Figure 1.11B: Settings to enable and manage memories Claude has about you

3. You can control if memories are generated from chat conversations.
4. You can delete a memory entry using the trashcan button next to a specific memory.
5. You can add a memory by typing text in the bottom box and sending it.

Claude will reference earlier conversations automatically, picking up patterns over time even when you have not asked it to remember anything specific. You control both behaviors in your personalization settings.



Good practice: Review your saved memories every few months, the same way you would review any list that grows without you noticing. Deleting a conversation does not delete the memories it created, so remove those separately if you want them gone.

Use incognito chats for isolated work

Sometimes you want a conversation that leaves no trace in memory, such as when you are testing a sensitive question, helping someone else for a single session, or discussing something you would rather keep separate from your usual profile. An **incognito chat** does not use your saved memory, does not get added to it, and does not appear in your conversation history afterward.

Start one from the model or chat menu and choose **Incognito chat**. Use it whenever you want a clean, one-off conversation that will not shape how Claude responds to you later.

Create a project with instructions and reference material

A **project** is a workspace that groups related chats, files, and instructions around one continuing piece of work. Instead of starting fresh each time you return to a task, you keep everything in one place.

1. In the sidebar, select **Projects**.
2. In the **Projects** page, select the **New project** button.
3. In the **Create a project** dialog box, give it a clear name, such as **Weekly Team Report** or **Learning Spanish**, type a description of the project, and then click the **Create project** button:

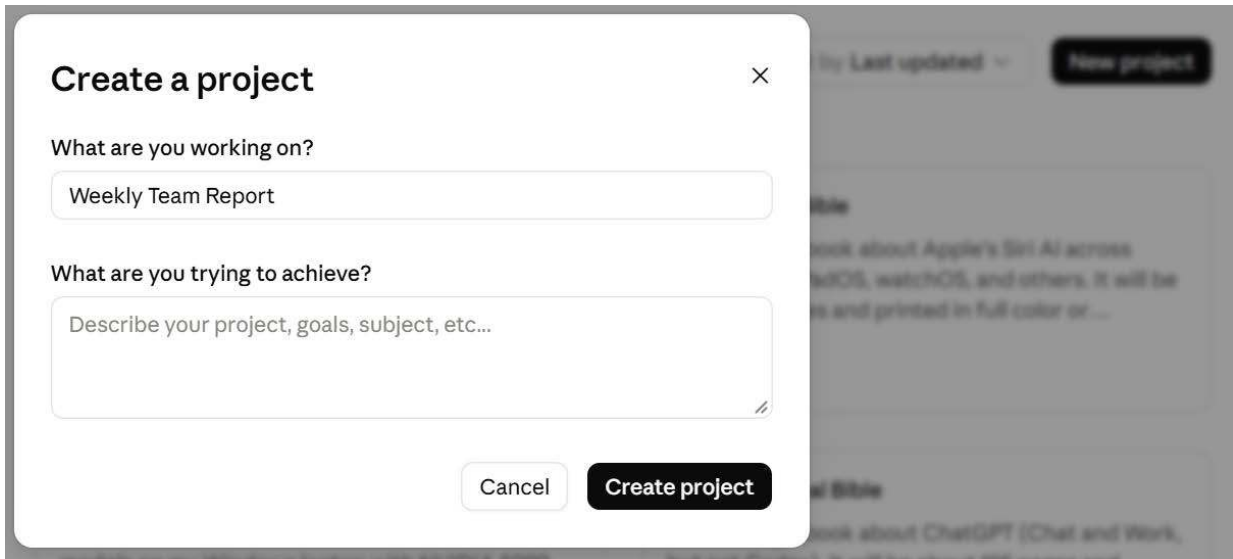


Figure 1.11C: Creating a new project with a name and description

4. On the project page, click the ... button and note that you can edit details like name and description, or delete the project:

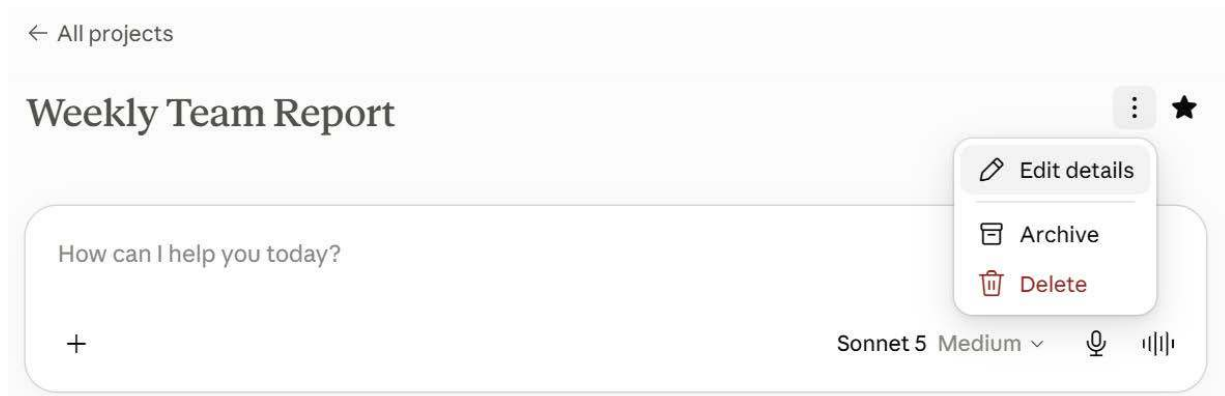


Figure 1.11D: Project settings menu

5. Scroll down the project page, and note that you can edit instructions and add files, such as a template, a style guide, or background material:

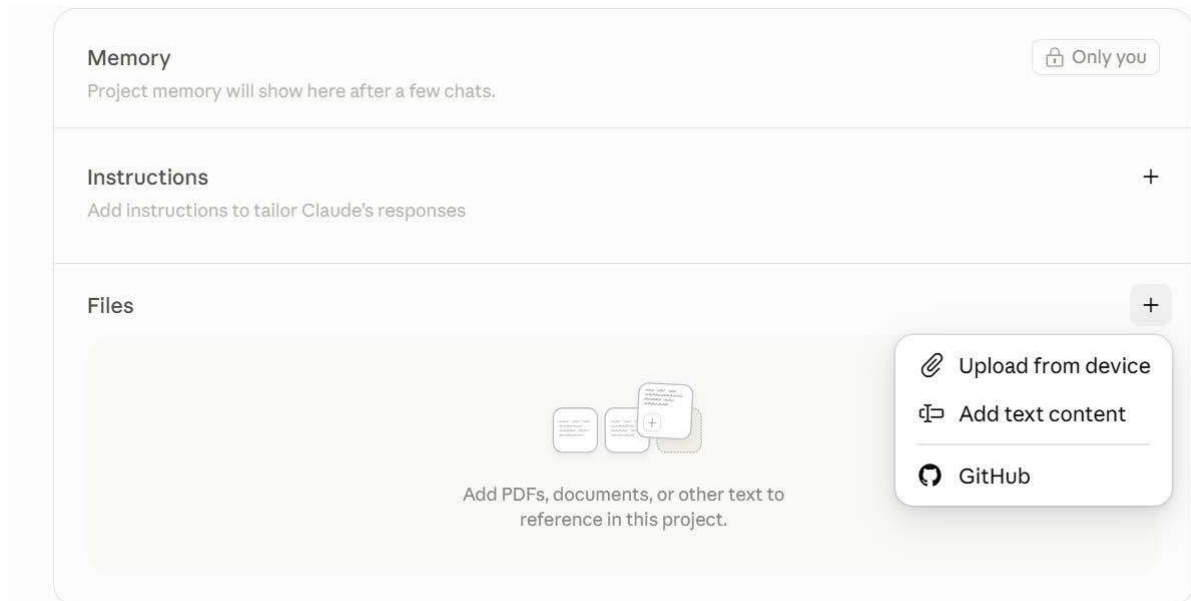


Figure 1.11E: Everything this project needs, including instructions, files, and memory of past conversations, stays together

6. Start a conversation inside the project whenever you work on this task.



Learn more online: *What are projects?* <https://support.claude.com/en/articles/9517075-what-are-projects>.

Memory from chats outside projects belongs to your general, non-project memory. Each project has its own separate memory, so information from your general chats does not automatically become part of a project's memory.

Try it now

Set one custom instruction about your role or preferences. Then create a project for a task you return to often, add a short instruction and one reference file, and start a conversation inside it.



What to ask next: List everything you currently remember about me, separated into custom instructions, general memory, and this project's own memory. Tell me if anything looks outdated.

Check the result

- ☐ Did a new conversation reflect your custom instructions without you restating them?
- ☐ Can you find, edit, and delete an item in your saved memories?
- ☐ Does your new project keep its instructions and files separate from your general chats?

The project you just created is not the finished workbench yet. *Chapter 1.12* brings everything from this Book into that project for one complete, working result.

Chapter 1.12 Project – Build Your Personal AI Workbench

You will bring together everything from *Book 1* into one finished result: a project set up around a task you actually do, ready to use again the next time that task comes around.

Before you begin, make sure you have worked through *Chapter 1.1* to *Chapter 1.11*. You will use prompting, research, file handling, and projects together in this chapter, not as separate skills. By the end, you will have a working project containing instructions, reference material, and one completed round of the task, ready to repeat the next time you need it.

Choose a task worth repeating

Pick something you expect to do again, not a one-time question. This book serves a wide range of readers, so choose whichever version fits your life right now:

Track	Good for	Example
Learn a subject	Students, career changers, curious learners	Building a study routine for a language or a certification exam
Investigate and decide	Anyone facing a recurring choice	Comparing options each time you plan a family trip
Complete a recurring assignment	Employed and independent workers	Writing a weekly status update or a client summary

Table 1.12: Tracks to choose from based on a task worth repeating

Whichever track you choose, the steps that follow work the same way. Write down, in one sentence, what a finished result looks like for this task. You will check your work against that sentence later.



Real-world example: A freelance bookkeeper chose the third track and built a project for preparing monthly client summaries. A retired teacher chose the first track and built a project for learning conversational Italian before a trip.

Create and configure the project

Now let's create and configure the project:

1. Create a new project and give it a name that describes the task, not only the subject, such as "Monthly Client Summary" rather than "Bookkeeping."
2. Edit the project's instructions and describe the task in the terms you used in *Chapter 1.5*: what you want produced, who it is for, and any constraints. Set the tone and format you want Claude to use every time, so you do not need to repeat it in each conversation:

```
This project is for preparing my monthly client summary. Each summary should be under one page, written for a non-technical client, and end with one recommended next step. Ask me for the month's figures if I have not provided them.
```

3. You should now see a project with a name and a short set of instructions saved inside it.

Add source material and instructions

Upload anything you will need every time you do this task: a template, a past example, a style guide, or background data.



Sample files: If you do not have your own files then you can download two sample files from the following link: <https://github.com/markjprice/claude-vb/tree/main/Files/Book1>

← All projects

Instructions

This project is for preparing my monthly client summary. Each summary should be under one page, written for a non-technical client, and end with one recommended next step. Ask me for the month's figures if I have not provided them.

✎

Files +

1% of project capacity used

1-12-Last-Month-Client-Summary-June-2026.docx

34 lines

DOCX

112ThisMonthFiguresJuly2026.xlsx

XLSX

Figure 1.12A: A project home screen with instructions and files, all in one place

If your task depends on current information, such as prices or availability, plan to search or run research inside the project each time, as you practiced in *Chapter 1.8* and *Chapter 1.9*, rather than relying only on saved files.

Complete, review, and repeat the workflow

Now run the task for real, inside the project:

1. Start a new conversation inside the project and ask for the finished result:

Using the figures in This-Month-Figures-July-2026.xlsx, prepare the July 2026 monthly client summary. Follow the project instructions and use last month's summary as the model. Update every figure, do not invent explanations beyond the notes in the workbook, and return the result as a Word document.

2. Note the response:

The screenshot shows the Claude AI interface. On the left, a sidebar contains icons for file management and a chat window. The main area displays a project workspace titled 'Monthly Client...' with a file 'July 2026 monthly client sum...'. A text box contains instructions: 'Using the figures in This-Month-Figures-July-2026.xlsx, prepare the July 2026 monthly client summary. Follow the project instructions and use last month's summary as the model. Update every figure, do not invent explanations beyond the notes in the workbook, and return the result as a Word document.' Below this, a message states 'Ran 5 commands, viewed 4 files, created a file - 2 notes'. A preview of the generated Word document is shown, titled 'NORTHSTAR HOME STUDIO Monthly Client Summary July 2026'. The document includes a table of results and a summary of what happened and what it means.

July 2026 client summary · DOCX Download as DOCX

Using the figures in This-Month-Figures-July-2026.xlsx, prepare the July 2026 monthly client summary. Follow the project instructions and use last month's summary as the model. Update every figure, do not invent explanations beyond the notes in the workbook, and return the result as a Word document.

Ran 5 commands, viewed 4 files, created a file - 2 notes

Here's the July 2026 summary for Northstar Home Studio, built from the workbook figure and following June's format.

Write a message...

Sonnet 5 Medium

Claude is AI and can make mistakes. Please double-check responses.

NORTHSTAR HOME STUDIO
Monthly Client Summary
July 2026

July was another strong month. Website traffic, leads, and sales revenue all grew, two new initiatives began paying off, and the cost of generating each lead fell further.

Results at a glance

Metric	July	June	Change
Website visits	10,680	18,420	+77%
New leads	612	526	+14%
Booked consultations	137	121	+13%
Sales	54	47	+15%
Sales revenue	\$82,800	\$71,800	+15%
Advertising spend	\$8,750	\$8,400	+4%
Cost per lead	\$14.30	\$15.67	9% lower

What happened

The summer search campaigns launched on July 7 and helped bring in more website visitors, lifting traffic from 18,420 to 10,680 visits.

On July 14, the mobile landing page was simplified and the inquiry button made more prominent. This helped more visitors make an inquiry, with new leads rising from 526 to 612.

The booking page improvement from last month continued to perform well, with booked consultations up from 121 to 137, and more of those consultations turned into completed sales, up from 47 to 54.

What it means

The campaign became more efficient again this month. Advertising spend rose by only 4%, while new leads rose by 14% and sales revenue rose by 15%. The cost of gaining each lead fell from \$15.67 to \$14.30.

Recommended next step: Keep the summer search campaigns funded at its current level, since it drove July's rise in traffic and leads, and roll out the same simplified layout used on the mobile landing page to other key pages on the site.

Figure 1.12B: Claude's response includes a downloadable Word document

3. Compare the result against the one-sentence description of success you wrote earlier.
4. Ask for specific revisions where the result falls short, using the targeted revision approach from Chapter 1.5.
5. Once you are satisfied, note anything you would change about your instructions or files for next time.

Check the result

- ☐ Does the output match the one-sentence description of success you wrote at the start?
- ☐ Did the project use your uploaded files and instructions without you repeating them?
- ☐ Is there a small change to your instructions that would save you time next time?

The next time this task comes around, start a new conversation inside the same project instead of beginning from nothing. Update the instructions or reference files whenever your needs change.

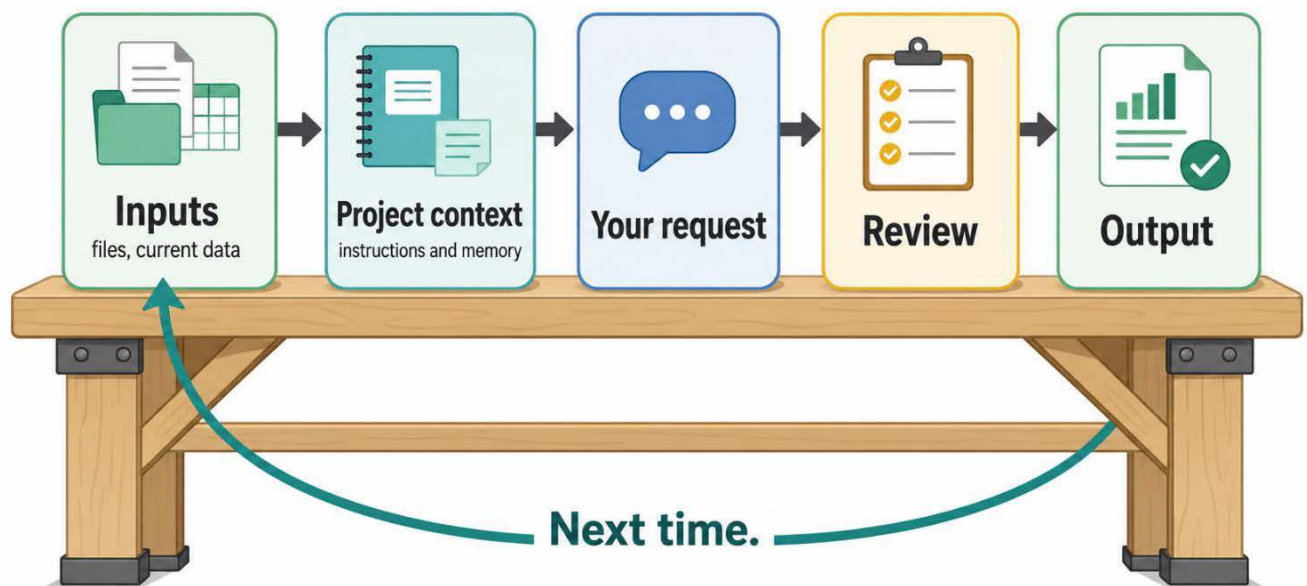


Figure 1.12C: Your finished workbench: a repeatable loop of inputs, context, request, review, and output, ready to run again whenever this task returns

You now have a working example of the pattern this book will keep returning to: a task, organized context, a clear request, and a human review before you rely on the result.

The next step

Book 2, Create Real Work Results, builds on this pattern with Claude Cowork, longer assignments, and creating more artifacts. Keep this project. You may extend it rather than replace it.

Book 2

Create Real Work Results

In *Book 2*, you interact with AI as a coworker to produce documents, spreadsheets, presentations, analysis, images and other media, and you learn to supervise longer, multi-step assignments.

It has nine chapters, including an end-of-book project:

- 2.1 Use Claude Safely and Responsibly at Work 70
- 2.2 Choose the Right Mode for Each Task 73
- 2.3 Draft Professional Documents 79
- 2.4 Summarize and Extract Information 86
- 2.5 Analyze Spreadsheets and Data 91
- 2.6 Create Visual and Interactive Content 98
- 2.7 Create Presentations and Visual Reports 104
- 2.8 Add Plugins, Connectors, and Skills 110
- 2.9 Project – Produce a Useful Weekly Briefing 118



Chapter 2.1 Use Claude Safely and Responsibly at Work

Book 1 showed you how to use Claude for personal questions and everyday planning. This chapter marks a change. From here on, many of your conversations will involve work: client information, company documents, or decisions that affect other people.

Before you draft a single report, you need a few ground rules. This chapter explains how to use Claude at work without putting your job, your employer, or your customers at risk. You don't need any special settings to follow it. You need your employer's policy and a habit of checking your work.

By the end of this chapter, you'll know what you can safely type or upload, when you need to say that AI helped you, and when a task needs a qualified professional instead of Claude.



Watch out: In *Book 2* we will only use **Cowork** features. Make sure you are in **Cowork** mode for all the chapters in *Book 2*. All chapters in *Book 2* require an individual paid plan or organization-managed workspace.

Follow your employer's AI policy

Many organizations now have a written policy that covers AI tools. It may list approved products, blocked types of content, and required disclosures.

Find your employer's policy before you use Claude for work tasks. Ask your manager or IT department if you can't find one. If no policy exists yet, treat the rules in this chapter as a reasonable default, and check with your manager before you handle sensitive material.

A policy can change. Review it again if your employer switches AI vendors, updates its terms, or announces new rules.



Watch out: If your organization's policy conflicts with anything in this chapter, follow the policy. It reflects decisions made for your specific workplace.

Know the difference between a personal account and an approved workspace

A personal Claude account belongs to you. An approved business workspace belongs to your employer and is managed by an administrator. The two can look similar, but they follow different rules for data storage, retention, and access.

Check which account you're signed into before you start work. Look at the account name or workspace label in the corner of the screen. If your employer hasn't set up a workspace, ask before you use your personal account for company material.

Some organizations allow personal accounts for low-risk tasks, such as brainstorming or learning, but require an approved workspace for anything involving customer or company data. Follow your organization's rule, not a general guess.



Learn more online: Anthropic’s support article *Move your personal Claude account to a Team or Enterprise organization* explains how an Enterprise workspace differs from a personal account: <https://support.claude.com/en/articles/9267400-move-your-personal-claude-account-to-a-team-or-enterprise-organization>.

Keep confidential and regulated data out of Claude without permission

Claude can help with many tasks, but some information should never go into a prompt without your employer’s permission. This includes passwords, private keys, unpublished financial results, health records, personal data about customers or colleagues, and anything covered by a nondisclosure agreement.

One way to decide what’s safe to share is to sort information into three groups:

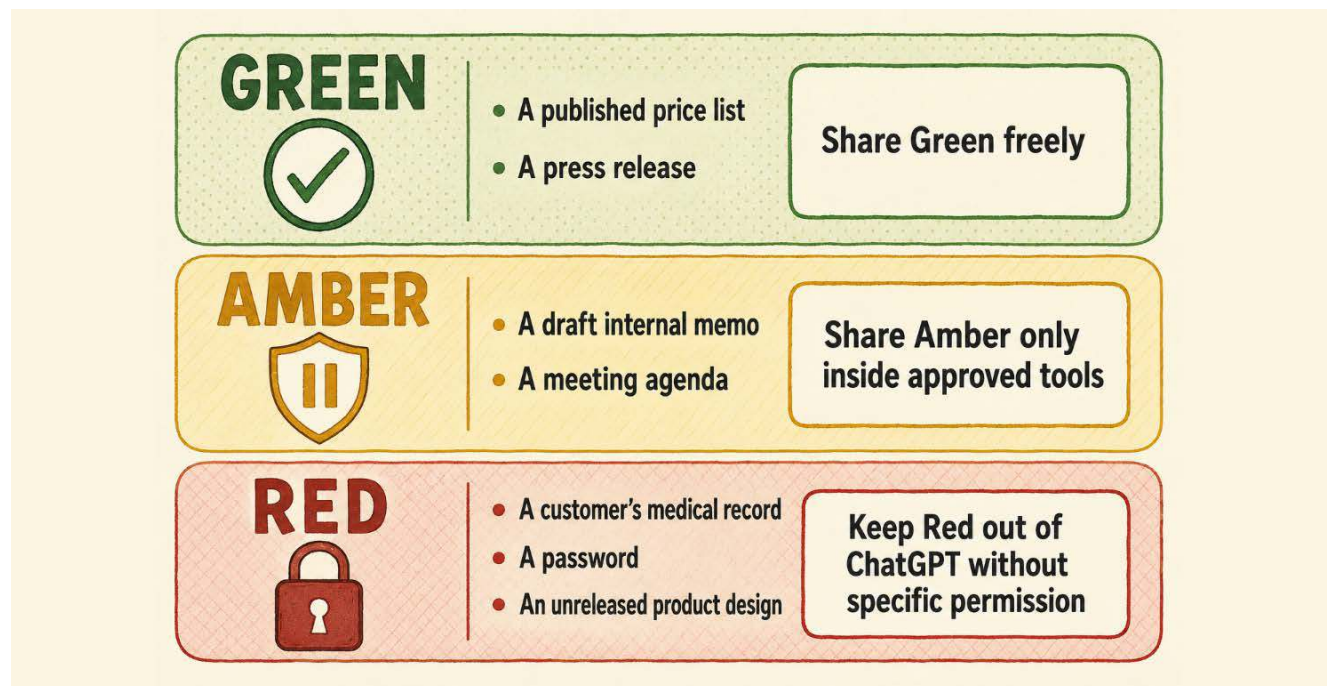


Figure 2.1A: Sort information into these three groups before you share it with Claude. When you’re not sure which group applies, treat it as red.

Treat red information as off-limits unless your employer has approved a workspace built to handle it. When in doubt, ask before you paste.



Real-world example: A marketing manager pastes a client’s campaign brief into Claude to draft social media captions. Before she sends the brief, she removes anything classed as red (client contact details and budget figures) and amber (specific client names), because her agency’s data classification treats these as confidential. She keeps the green data, like general messaging goals and target audience, which is safe to share.



Watch out: Pasting an email thread, a spreadsheet, or a screenshot can carry hidden information you didn’t intend to share, such as other people’s names, phone numbers, or comments. Check the full content before you upload it, not only the part you meant to use.

Verify facts, calculations, quotations, and citations

Claude can produce a confident-sounding answer that contains an error. This can happen with a date, a statistic, a quotation, or a calculation. Treat every factual claim as a draft until you check it.

Compare important numbers against your source spreadsheet or system of record. Look up quotations in the original document. Recalculate totals that will appear in a report or an invoice. This check is most important for material other people will rely on, such as a client proposal or a board report.



Good practice: Ask Claude to show its sources or its working before you accept a number or a quotation. A response with no source is harder to check.

Disclose AI assistance when it's required

Some organizations, clients, or professional bodies require you to say when AI helped produce a piece of work. A contract, a regulator, a journal, or a school may set its own rule.

Check your employer's policy, your contract, and any relevant professional or industry rules before you submit AI-assisted work. If you're unsure, ask rather than guess. Disclosure requirements are becoming more common.

Retain human approval for consequential actions

Claude can now do more than answer a question. Cowork and connected tools can send messages, edit files, and complete tasks on your behalf. That capability makes a review step necessary before anything happens that can't easily be undone.



Figure 2.1B: Claude's output is one step in a process you still control. Pause at an approval point before any action that can't easily be reversed.

Before you let Claude act on your behalf, check what it plans to do, and approve it. Require this step for any action that sends a message, publishes content, spends money, shares data, or changes a system that other people depend on.

Know when a qualified person needs to be involved

Claude can help you understand a medical, legal, financial, employment, or safety question, but it can't replace a licensed doctor, lawyer, accountant, HR professional, or safety officer. These fields carry consequences that go beyond an incorrect document, such as a missed diagnosis, a broken contract, or an unsafe decision.

Use Claude to prepare for these conversations. Ask it to help you list questions, organize your situation, or understand unfamiliar terms. Then bring your findings to the right professional before you act.

Try it now

Find your employer's written AI policy or ask your manager whether one exists. Note three things it allows and three things it doesn't. If no policy exists, write down which rules from this chapter you'll follow by default, and confirm your list with your manager.

Check the result

- ☐ Can you explain, in one sentence, what your organization treats as confidential?
- ☐ Do you know who reviews AI-assisted work before it reaches a client or the public?
- ☐ Have you identified one task in your role where a qualified professional, not Claude, should make the final call?

You now know what you can safely share with Claude at work and when to bring in a qualified professional instead. The next question is where to have that conversation. Claude offers three modes built for different kinds of work, and picking the right one before you start will save you time throughout the rest of this book.

Chapter 2.2 Choose the Right Mode for Each Task

Claude brings three modes together in one application: **Chat**, **Cowork**, and **Code**. Each mode suits a different kind of task. Picking the right one before you start saves time and produces a better result.

This chapter will help you decide which mode to open. You should already be comfortable having a basic conversation with Chat. By the end, you'll be able to look at a task and know within seconds whether it belongs in Chat, Cowork, or Code.

Use Chat for questions, focused help, and single-step research

Chat is the mode most readers already know. Choose it when you:

- Have a single question
- Want to talk through an idea
- Need a short piece of writing, an image, or a quick analysis

Chat works well for tasks you can finish in one sitting, perhaps by adding a few files. Asking for an explanation, comparing two products, drafting a short email, or improving a paragraph all belong here.

Chat responds quickly and keeps the conversation in view, so you can revise your request through a few back-and-forth messages. Use it when you expect to stay closely involved in every step.

Use Cowork for multi-step research and finished deliverables

Cowork runs across desktop, web, and mobile, although some local-file, local-connector, live-artifact, and computer-use capabilities still require desktop.

Cowork is built for larger assignments. Choose it when a task:

- Needs research across several sources and steps
- Needs to read from or write to a collection of files in a local folder
- Should end in a finished image, document, spreadsheet, presentation, or report
- Will take more than a few minutes of back-and-forth to finish in Chat

Cowork can plan a multi-step assignment, carry it out over several minutes, and show you its progress along the way. You can step away and check back, rather than watching every message appear, or get on with other work until you get a notification that it's finished.

Personally, I distinguish Chat and Cowork mainly by interaction style and level of delegation rather than by whether they use more advanced features like connecting external services. One of the most important and useful features of Cowork that Chat doesn't have is folders.

Working with folders

Uploading a few files works well when Claude only needs them for one task. But sometimes the material you want Claude to work with already lives together in a folder on your computer. In Cowork on the desktop, you can select that folder and give Claude direct access to the files inside it.

A folder is especially useful when a task involves several related files, or when Claude needs to create and update files as it works. For example, imagine you have a folder for a client project containing meeting notes, a budget spreadsheet, a project brief, previous reports, and images. Rather than uploading those files one at a time, you can give Cowork access to the project folder and ask Claude to work with the material there.

You might select a folder when you want Claude to:

- Review several files that belong to the same piece of work
- Find information without you knowing exactly which file contains it
- Create new documents alongside the source material
- Update or reorganize files already stored there
- Return to the same collection of files for related tasks

To select a folder for a conversation, make sure you are in **Cowork** mode, and then under the chat box, select **Project or folder**:

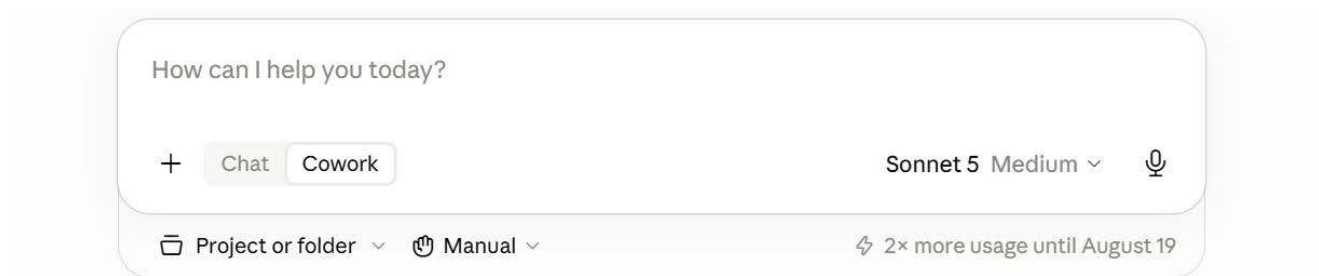


Figure 2.2A: Cowork mode in the chat box and its Project or folder control

Then you can enter a prompt that references the folder you selected:

```
Work with the files in this folder for the Greenfield website redesign. Read the
project brief, meeting notes, and budget before doing anything else. Then create a
one-page status report summarizing progress, open decisions, risks, and actions due
this week. Save the finished report in this folder.
```

Giving Claude access to a folder is different from uploading a copy of a file. Cowork can read files directly from the folder and, when the task requires it, write files back there. This makes folders useful for work that changes over time rather than a single question about a fixed document.

If you expect to return to the same work repeatedly, consider creating a Cowork project around the folder instead of selecting it only for one task. This saves you from rebuilding the same working environment every time you return to the project. You will learn about projects in *Chapter 2.8*.

Give Claude the smallest useful folder rather than a broad folder such as **Documents** or your entire home directory. Anthropic recommends creating a dedicated working folder instead of granting unnecessarily broad file access. Claude can read and write within folders you connect, so check what the folder contains before granting access and keep backups of important files. Treat folders as a sandbox that limits what Cowork can access.



Watch out: Selecting a folder does not mean that every file inside it is relevant to the current task. Tell Claude which files or types of information it should use and tell it what to ignore. A clearly named project folder containing only relevant material makes this much easier.



What to ask next: Before you start, list the files in this folder that you think are relevant to my task and explain briefly how you plan to use each one.

Recognize when a task has grown too big for Chat

Some tasks start small and grow. A single question about a competitor turns into a request to compare five companies across ten criteria, using data from three files. At that point, Chat is no longer the right tool.

Watch for these signs that a task belongs in Cowork instead:

- You've asked several related questions in a row and are stitching the answers together yourself
- You keep re-explaining the same background information
- You need Claude to check multiple sources before it answers
- The result should be a shareable file, not a chat message

You don't have to start over. Most Cowork assignments begin with the same plain description of what you want that you'd type into Chat. The difference is what happens after you send it.

Starting a large, multi-source assignment in Chat can leave you copying and pasting between messages for far longer than the task should take. If a task keeps growing while you work on it, stop.

If you're working inside a project, switch to Cowork in that same project and start a fresh Cowork chat so your files and context carry over and you're not starting from zero. If you're not in a project, start a new project first, or move the chat into an existing project rather than continuing in Chat.

Know when a task belongs in Claude Code instead

Claude Code is the mode for building and changing software. Choose it when your task involves writing or editing code, working with a repository, or running tests.

Figure 2.2A shows the mode selectors in the Claude desktop app:

1. **Chat:** For questions and focused help
2. **Cowork:** For research and finished deliverable artifacts

3. Code: For building and changing software



Figure 2.2B: The mode selectors in the Claude desktop app

Code is Claude’s specialized software-development environment. You can use it on the web, in the Claude mobile and desktop apps, in a terminal, and through supported IDE integrations. In the desktop app, **Home** and **Code** sit as two separate top-level views: you pick **Home** or **Code** from the top-left menu, and **Chat** and **Cowork** only live inside the Claude chat box as a toggle.

This book does not cover Code at all. If you want to learn how to use Claude to build websites or mobile apps, the separate book *Claude Code Visual Bible* covers that software development work.

Dispatch is a Cowork capability for remotely interacting with Claude and the user’s desktop environment. Dispatch runs your tasks on your desktop, so your computer needs to be awake and the Claude desktop app open. This is different from a cloud-hosted session, which runs on Anthropic’s servers and keeps working even when your computer is off. At the time of writing, Dispatch is still a beta feature so check the online article about it for the latest information: <https://support.claude.com/en/articles/13947068-assign-tasks-from-anywhere-in-claude-cowork>.



Good practice: If you’re not sure which mode fits, describe the task to Chat first and ask it whether the assignment would be better suited to Cowork or Code. A short check like this costs little time and can save you from restarting a large task in the wrong place.

Complete multi-step assignments and supervise computer actions

Cowork is intended for longer assignments involving research, connected files and apps, and finished documents, spreadsheets, presentations, and reports. This section covers defining the deliverable, letting Cowork plan and execute the assignment, staying involved along the way, and reviewing what it actually completed.

The common seven steps for longer assignments should include:

1. **Define the final deliverable:** State what you want at the end: a specific document, spreadsheet, presentation, or report, not just a general area to investigate. A clear deliverable gives Cowork something concrete to plan toward.
2. **Let Cowork create and execute a plan:** Once you describe the assignment, get Cowork to propose a plan before carrying it out. Read the plan before approving it. This is your chance to catch a misunderstanding before Cowork spends time on the wrong task.
3. **Answer questions and redirect progress:** Cowork may ask you a question partway through, or you may notice it heading in the wrong direction. Answer promptly and redirect clearly rather than waiting until the end to fix a problem that grew along the way.



Real-world example: A consultant hands Cowork a request to compare three vendors and produce a one-page recommendation, then reviews each research step before approving the final document rather than reading only the finished summary.

4. **Distinguish assistance from autonomous action:** Some steps only produce a draft for you to review. Others, such as sending a message or changing a shared file, act outside the chat and can affect other people. *Figure 2.2C* shows this range as steps, from a suggestion you review yourself to an action taken on your behalf:

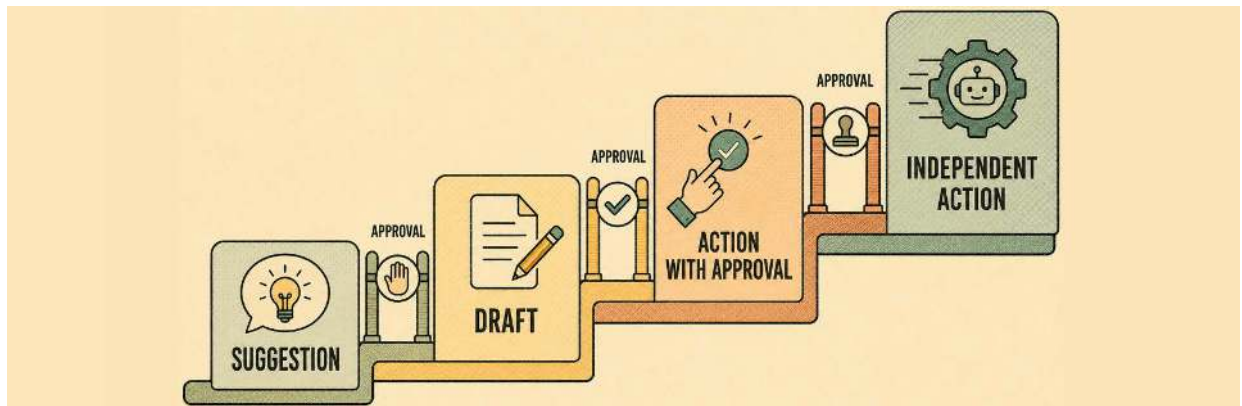


Figure 2.2C: A suggestion and an action carry very different levels of risk

5. **Review logs, permissions, and sensitive data:** Before approving a step that needs a login or access to a file, check what permission it's requesting and whether that permission matches the task. Do not grant access to more accounts or data than the assignment needs.
6. **Require approval before consequential actions:** *Figure 2.2D* traces a full assignment from goal to deliverable, with review built in at each stage:

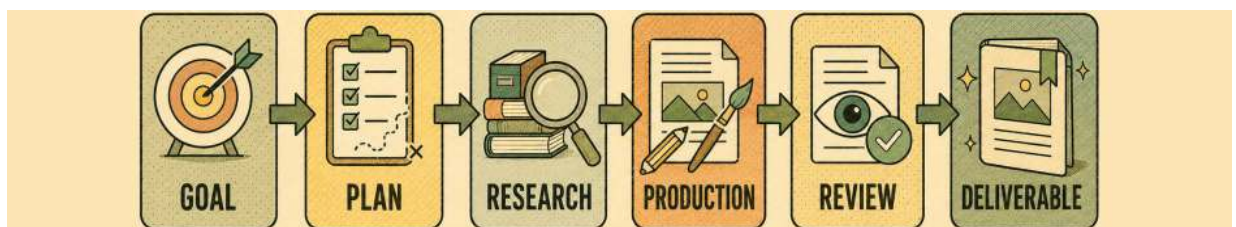


Figure 2.2D: Review sits before the deliverable, not after. Catching a problem at this stage is easier than fixing a finished document.



Watch out: An action that sends a message, publishes content, spends money, or changes shared data cannot always be undone. Require your own approval before any such step, even if Cowork presents it as routine.

7. **Check what the system actually completed:** When Cowork reports that it finished an assignment, open the actual deliverable and compare it against your original request, rather than trusting the summary of what it did. A report that says a step succeeded is not the same as confirming the result is correct.



Good practice: Keep a short record of what you approved during a long assignment. If a result later looks wrong, this record helps you find which step introduced the problem.

Know when a Cowork assignment needs you back

A Cowork assignment can run for several minutes without you watching every step, so the Claude desktop app provides notifications.

1. In **Settings**, select **General**.
2. In **Notifications**, switch **Response completions** on.
3. Separate switches control notifications from Claude Code via email, and push notifications on your phone:

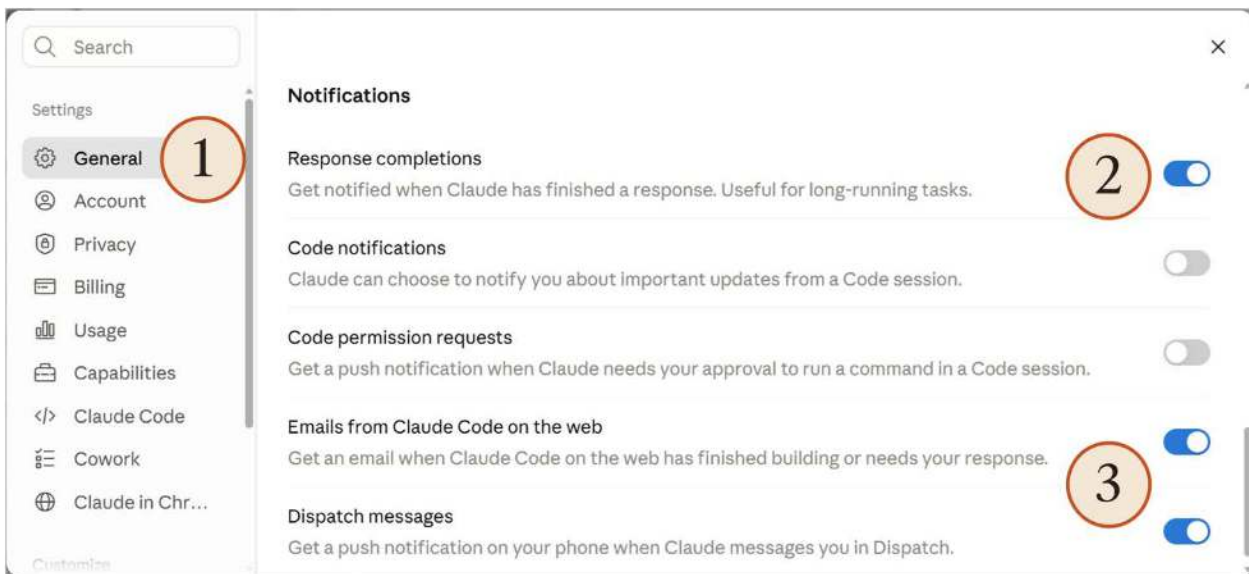


Figure 2.2E: Notification settings

Your operating system may also ask permission to let the Claude desktop app show notifications; grant it if you want an alert outside the app window.



Watch out: A notification banner can reveal part of an assignment, such as a document title or a step Cowork just completed, to anyone glancing at your screen. Treat this the same as any other Cowork output: keep alerts to permission and question notifications only if you're handling anything confidential or working on a shared screen.

Try it now

In a current conversation, find the model selector and the reasoning-effort control. Pick a task you'd normally split across several Chat messages, such as researching three options for a purchase and writing up a short comparison. Open Cowork and describe the whole task in one message: what you want to compare, what sources are acceptable, and what the final comparison should include. Watch Cowork outline its plan before it begins.



What to ask next: Based on what I just described, would this task work better in Chat or Cowork?

Check the result

- ☐ Can you locate the model and reasoning effort controls without help?
- ☐ Does the plan Cowork proposed match what you actually asked for?
- ☐ Are the sources it used ones you'd trust for this decision?
- ☐ Is the finished deliverable something you could hand to someone else without further editing?



Learn more online: *Get started with Claude Cowork:* <https://support.claude.com/en/articles/13345190-get-started-with-claude-cowork>.

Once you know which mode fits your task, you're ready to put it to work. *Chapter 2.3* starts with one of the most common jobs you'll hand to Claude: turning a blank page into a finished professional document.

Chapter 2.3 Draft Professional Documents

Claude can produce a full first draft of a report, proposal, letter, or policy in minutes. But the draft is only as useful as the plan behind it, and a first draft is rarely the final version.

This chapter walks through seven steps: define the reader and purpose in a brief, build an outline, generate a complete draft, get useful feedback, keep the document sounding like you, check its claims, and finally produce one controlled final version and move it into the format you need:

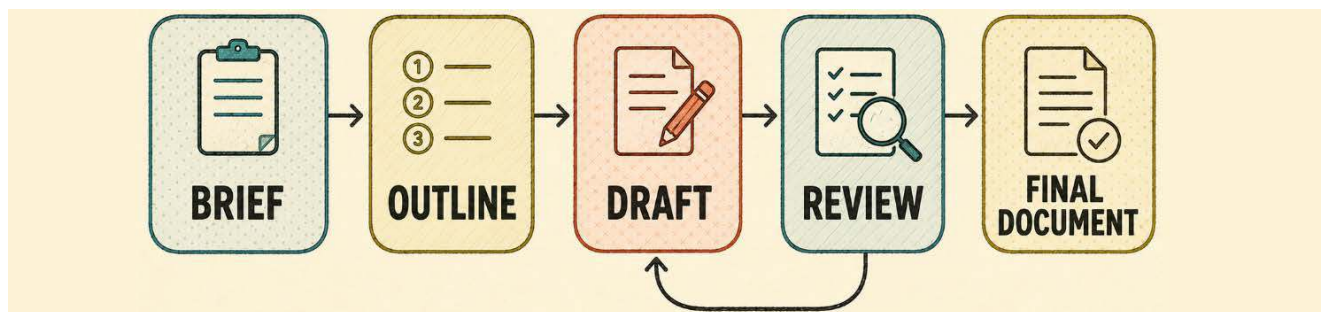


Figure 2.3A: Treat the outline as a checkpoint. Catching a structural problem here is faster than catching it in the final draft.

You need a clear idea of the document you want to produce and a rough sense of its audience. By the end of this chapter, you'll be able to turn a document request into a finished draft ready for review.

Define the reader and purpose

Before you ask Claude to write anything, decide who will read the document and what you want that reader to do after reading it. A status update for your manager calls for a different length and tone than a proposal for a client or a policy for new staff.

Tell Claude the reader, the purpose, and any constraints you already know, such as a required length, format, or deadline. A short brief like this does more to improve the draft than a longer, vaguer request.

Start by asking Claude to help you define the document before writing it:

```
I need to write a short professional report about [project or topic].
```

```
Help me define:
```

- Who will read it
- What the reader needs to know
- What I want the reader to do after reading
- The required length, tone, and deadline

```
Ask me no more than five short questions. Then turn my answers into a brief I can use to plan the report. Do not write the report yet.
```



Sample files: If you don't have a real short professional report idea ready to turn into a brief yet, download <https://github.com/markjprice/claude-vb/tree/main/Files/Book2/2-3-sample-brief.docx>. It contains a short, filled-in project brief you can add to your chat for the following sections.



Good practice: State the reader and purpose in one or two sentences before you describe the content. For example: "This is a one-page update for my regional manager. She wants to know whether the project is on schedule and what she needs to approve this week."

Build an outline before drafting

Ask Claude for an outline before asking for the full document. An outline is quicker to check than a complete draft. Fixing a missing section or a wrong order is easier now than after the document is written:

```
Using the attached report brief, create an outline for the report.
```

```
Use no more than six sections. For each section, give:
```

- A short heading
- One sentence explaining what the section will cover

```
Put the information in the order that will be most useful to the reader. Do not write the report yet. If the brief is missing something important, tell me instead of inventing it.
```

Review the outline against three questions:

1. Does it cover everything the reader needs?
2. Does it include anything the reader doesn't need?
3. Does the order make sense?

Ask for changes before moving on.

Generate the first complete draft

Once the outline looks right, ask Claude to write the full document from it. Ask for one complete first draft rather than one section at a time. This keeps the tone and level of detail consistent from start to finish:

Using the attached report brief and the outline we agreed on, write the complete first draft.

Keep it to one page. Use plain, professional language, short headings, and bullet points where they make the report easier to scan. Use only facts from the brief. If any necessary fact is missing, mark it [VERIFY] instead of inventing it.

Create the report in an artifact so I can review and export it.



Officially, Anthropic distinguishes artifacts from generated downloadable files. Claude can directly create DOCX, XLSX, PPTX, and PDF files, while artifacts are standalone content, apps, tools, or visualizations displayed in a dedicated pane.

Figure 2.3B shows a response with a document in an artifact:

1. A dropdown menu has more options for working with the chat conversation: pinning it, marking it as unread, renaming it, adding it to a project, and deleting it.
2. Claude shows the steps it went through to generate the response and its artifact.
3. The artifact can be opened in Google Drive or downloaded to your local computer.



Real-world example: A nonprofit program officer drafts grant reports in Claude by giving it the funder’s required sections and word count up front, then checking every dollar figure against the program’s own budget spreadsheet before submitting.



Watch out: A fluent draft can still contain wrong figures, invented sources, or overstated claims. Confirm any number, date, or quotation before it goes to a reader who will act on it.

Ask for structural rather than vague feedback

A request such as “make this better” produces a vague response. Ask instead about specific parts of the document: the argument, the order of sections, the opening, or the ending.

Name what you want checked. Ask whether the document makes its point in the first paragraph. Ask whether each section follows logically from the one before it. Ask whether the conclusion matches the evidence in the body. These questions produce feedback you can act on.

There are four kinds of review you can ask for one at a time: accuracy, structure, tone, and presentation:

1. **Accuracy:** whether the facts, figures, dates, and claims are correct. Ask, for example, “Does every dollar figure in this proposal match the totals in the budget table?”
2. **Structure:** whether the sections follow a logical order and support the document’s main point. Ask, for example, “Does the evidence in section two support the conclusion in section three?”
3. **Tone:** how the document will sound to the person reading it. Ask, for example, “Does this message sound confident without sounding demanding?”
4. **Presentation:** whether headings, spacing, and formatting stay consistent throughout. Ask, for example, “Are the heading levels and bullet styles the same in every section?”

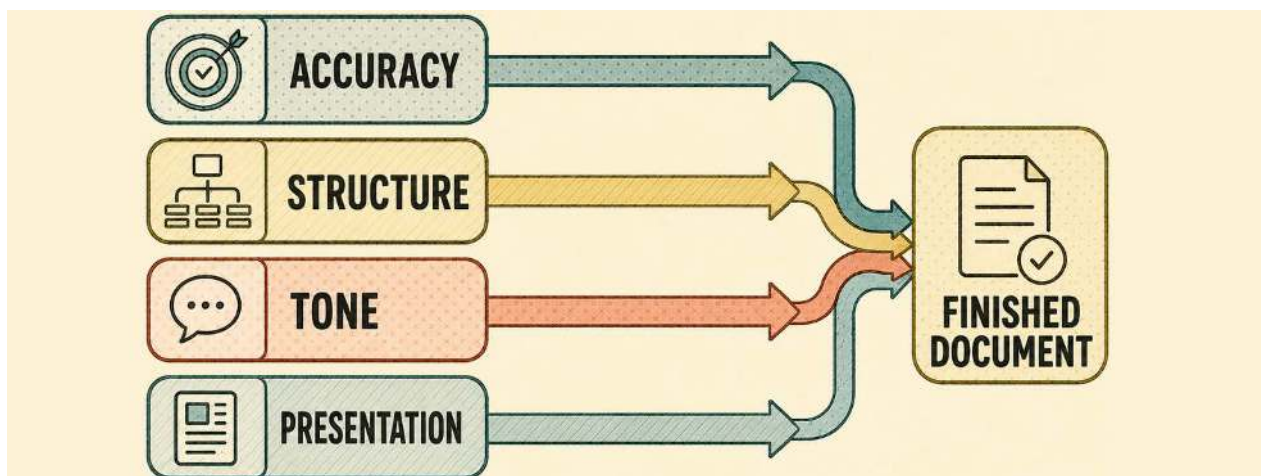


Figure 2.3D: Ask for one kind of review at a time. Mixing all four into a single request makes the feedback harder to act on.



Good practice: Ask one structural question per message. Reviewing the argument, then the section order, then the opening gives you three focused answers instead of one crowded one.

Preserve the intended voice

A revision should still sound like you or like the person the document represents. Tell Claude which parts of the tone and wording to keep before asking for changes.

If a revision changes more than you wanted, point out what changed. Ask Claude to restore it while keeping the rest of the improvement. This is faster than starting the revision over.



Real-world example: A small-business owner asks Claude to review a customer email for tone before sending it, specifically checking whether it sounds apologetic without sounding defensive.



Watch out: A revision aimed at clarity can shift the tone from formal to casual without you noticing, or the other way around. Read the revised version aloud, or have a colleague read it, before you send it.

Check claims and internal consistency

Before you accept a revision, check every fact, figure, date, and quotation against a source you trust. Claude can write confidently about a claim that is wrong, so a fluent sentence is not proof that the sentence is accurate.

Also check the document against itself. A number given in one section should match the same number used later. A term defined early in the document should keep the same meaning throughout.

Export or transfer the finished content

Once you have accuracy, structure, tone, and presentation feedback, ask for one final revision that applies the changes you approved. Naming exactly which changes to apply keeps Claude from introducing new, unreviewed edits along the way.

A finished draft still needs to reach its destination: a shared document, an email, a slide deck, or a printed page. Copy the finished text into your usual word processor. Or use the download option if Claude offers a direct download in the format you need. Check the formatting after moving the document, since headings, lists, and tables don't always carry over exactly.

Read the final version in full before sending it. A controlled revision still deserves the same check as the first draft.

Know what an Office add-in can see and do

Word, Excel, PowerPoint, and Outlook each have an official Claude add-in that opens a Claude panel inside the file you already have open, rather than sending you to `claude.ai` to upload a copy. *Chapters 2.3, 2.5, and 2.7* introduce each one in its own context. This section covers what they have in common before you meet them chapter by chapter.

The Claude for Microsoft 365 add-in signs in with the same Claude account you use everywhere else, so it follows your existing plan, workspace, and administrator settings rather than opening a separate trial. Once you're signed in, the add-in can read the content of the open file and the text you type into its panel, and it uses both as the basis for its response:

1. After installing the **Claude for Microsoft 365** add-in, select the **Claude** button in the toolbar to show the **Claude** panel.
2. Type your prompts into the chat box as usual:

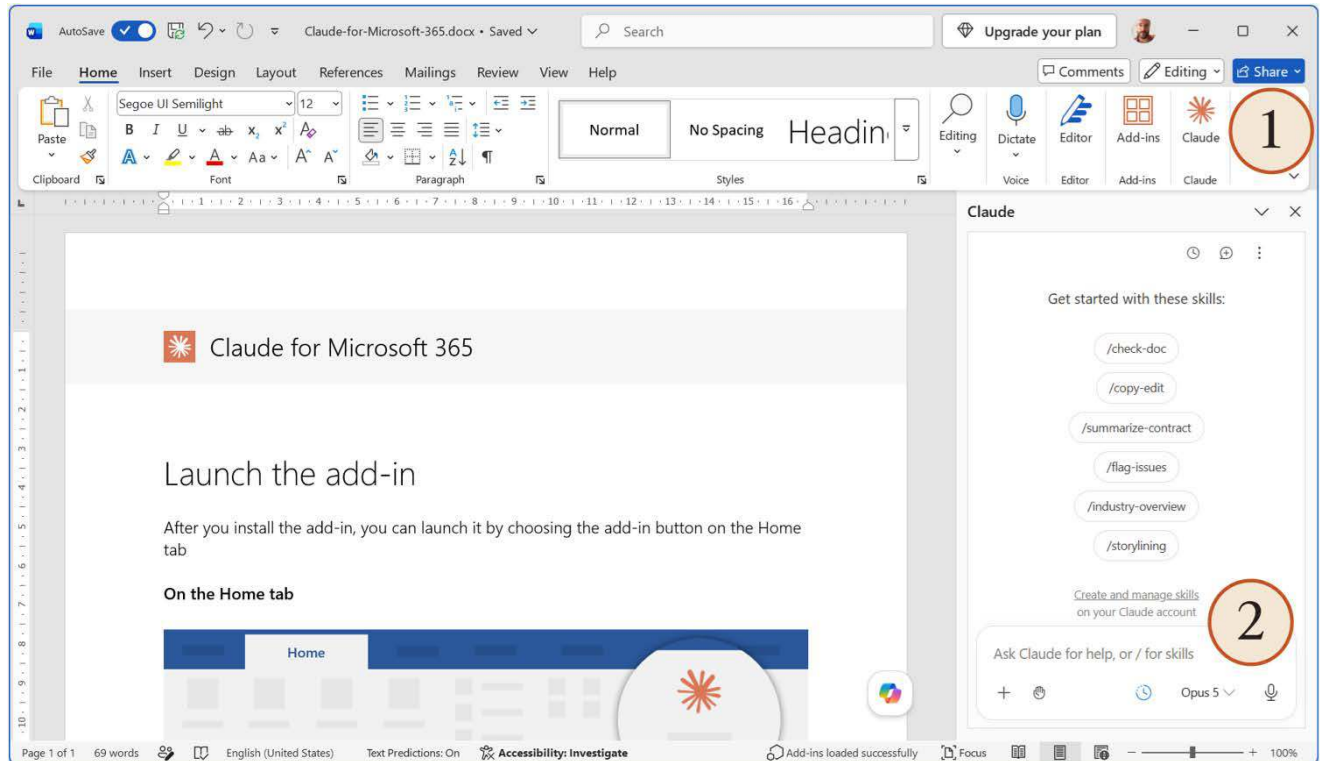


Figure 2.3E: Claude for Microsoft 365 extension in Microsoft Word

An Office add-in has limitations worth knowing before you rely on it. Its conversation runs separately from your regular Claude chat history and memory, so it can't recall an earlier conversation from the main app. But enabled Microsoft 365 add-ins can share context with each other through cross-app mode supporting one conversation spanning Word, Excel, PowerPoint, and Outlook. It works with the file that's open, not with files you haven't loaded. And some advanced, application-specific features, such as Excel macros and VBA, or PowerPoint's finer formatting and animation controls, may not be fully supported yet.

Treat an Office add-in the same way you'd treat Claude in a browser tab. Everything earlier in this chapter still applies: confidential material still needs your employer's permission before it goes into the panel, and any change the extension makes to your document, workbook, or deck is a draft that needs your review, not a finished result.



Learn more online: Use Claude for Word: <https://claude.com/docs/office-agents/word>.
 Install Claude for Microsoft 365: <https://marketplace.microsoft.com/en-us/product/office/WA200010725>.

Try it now

Choose a short document you actually need, such as a one-page project update or a short proposal. Tell Claude the reader and purpose, ask for an outline, review it, then ask for the full draft. Revise one section using a targeted instruction rather than requesting a new draft.

Or take a draft you already have. Ask Claude to review its structure only: does the argument hold together, and does each section follow from the one before it. Then ask a separate question about tone. Compare the two sets of feedback before deciding what to change.



What to ask next: List every change you made in the final revision. Confirm each one against my approved feedback.

Check the result

- ☐ Does the draft match the reader and purpose you described?
- ☐ Can you confirm every fact, figure, and quotation in the draft?
- ☐ Does the document read as one consistent piece, rather than several different voices stitched together?
- ☐ Does the feedback address the specific question you asked, rather than the whole document at once?
- ☐ Can you confirm every fact, figure, and quotation that changed during revision?
- ☐ Does the revised document still sound like the intended voice?



Learn more online: What are artifacts and how do I use them? <https://support.claude.com/en/articles/9487310-what-are-artifacts-and-how-do-i-use-them>.

Revising a document you already have is one skill. Making sense of documents other people wrote is another. *Chapter 2.4* turns to the meeting notes, email threads, and reports piling up in your inbox, and shows you how to pull out exactly what you need.

Chapter 2.4 Summarize and Extract Information

Meeting notes, long email threads, and reports often bury the few facts you actually need. This chapter covers four steps: summarize for a specific reader, pull out actions and dates, compare several documents, and keep a link back to the source behind every claim.

You need one or more documents, transcripts, or sets of notes to work with. By the end of this chapter, you'll be able to turn a pile of unstructured material into a short, structured result you can act on.



Sample files: If you don't have your own set of notes to work with, download this document: <https://github.com/markjprice/claude-vb/tree/main/Files/Book2/2-4-sample-meeting-notes.docx>. It contains three short sets of fictional meeting notes with a few actions and dates buried in each.

Summarize for a stated audience

A summary written for your own reference looks different from one written for a manager who wasn't in the meeting. Tell Claude who will read the summary and what they need to know before asking for one. Ask for the length you want, such as three bullet points or one short paragraph. A summary that runs longer than the reader has time for defeats its own purpose:

Using the attached meeting notes, write a summary for a manager who did not attend the meeting.

The manager needs to know:

- What was decided
- What action is required next
- Any deadline or risk that needs attention

Use exactly three concise bullet points. Lead with the information that requires the manager's attention. Use only facts stated in the notes. If something is unclear or missing, say so instead of guessing.



Good practice: Name the reader's next action. A summary for someone who needs to approve a budget should lead with the number they need to approve, not with background context.

Extract actions, decisions, dates, and risks

Meeting notes and long threads usually contain a few items that matter more than the rest: who agreed to do what, by when, and what could go wrong. Ask Claude to pull these into a short, structured list instead of a narrative summary:

Using the attached meeting notes, extract the information that people need to act on.

Create four sections:

1. Actions
2. Decisions
3. Important dates
4. Risks

For each action, include the task, owner, deadline, and status. Sort the actions by owner.

Use only information stated in the notes. Do not infer an owner, date, decision, or risk. Write "Not stated" when information is missing, and flag anything unclear for review. Combine duplicate items that appear in more than one set of notes.

Figure 2.4A shows a page of meeting notes turned into a table of actions, owners, and deadlines:

1. Actions

Owner	Task	Deadline	Status	Review flag
Jordan Miles	Train the support team.	August 6, 2026	Scheduled	None
Leo Chen	Investigate the email queue and report the findings.	August 3, 2026	In progress	The cause of the email delay is not confirmed.

Figure 2.4A: Turning notes into a table makes it easy to check that every action has an owner and a date

Figure 2.4B shows the page of meeting notes with decisions extracted:

2. Decisions

- Keep the August 14 launch target for now.
- The password-reset process must be fixed and retested by August 4 for the August 14 launch to remain realistic.
- Ask Maya Patel to approve up to US\$750 for extra vendor troubleshooting by August 1.
- Hold the final launch decision meeting on August 7.

Figure 2.4B: Turning notes into a bullet list makes it easy to validate every decision

Figure 2.4C shows the page of meeting notes with important dates extracted:

3. Important dates

Date	Event or deadline	
August 1, 2026	Deadline to ask Maya Patel to approve up to US\$750 for extra vendor troubleshooting.	
August 3, 2026	Leo Chen to report the findings from his email-queue investigation.	

Figure 2.4C: Extracting important dates into a table makes it easy to know what's happening and when

Figure 2.4D shows the page of meeting notes with risks extracted:

4. Risks

- If password-reset emails are still delayed after August 4, the August 14 launch may need to move.
- Password-reset emails sometimes take up to 15 minutes to arrive. The vendor's email queue settings may be causing the delay, but the cause is not confirmed. **Cause needs review.**

Figure 2.4D: Turning notes into a bullet list makes it easy to highlight risks

Check the extracted tables and lists against the source material. An action without a clear owner, or a date that doesn't appear anywhere in the notes, is worth a second look before you rely on it.



Real-world example: A team lead pastes the notes from three planning meetings into Claude and asks for one table of open actions, sorted by owner, instead of rereading all three sets of notes before the next meeting.

Compare several documents

Scattered notes, emails, and reports can become one structured comparison. For example, a small business owner sourcing a new supplier could paste rough notes from three separate phone calls with printing companies into Claude and ask it to generate a single table comparing price, delivery time, and support terms side by side:

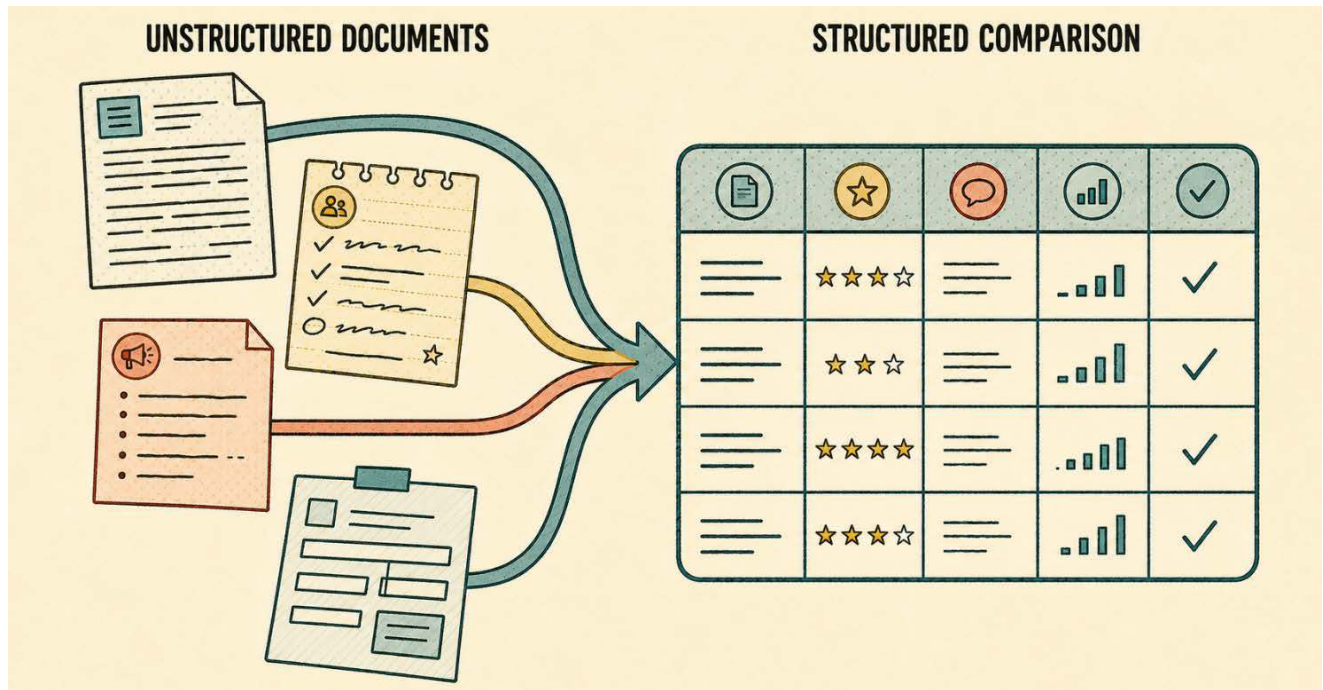


Figure 2.4E: Scattered notes, emails, and reports become one structured comparison

You may need to compare notes from several meetings, versions of a report, or proposals from different printing companies. Upload the documents together and ask Claude to compare them against the same set of questions.



Sample files: If you don't have your own set of files, download all the files with names that start with 2-4-compare from this folder: <https://github.com/markjprice/claude-vb/tree/main/Files/Book2/>. There are three fictional, one-page source documents in different formats: an email proposal, rough call notes, and a draft quote. Their facts overlap enough for comparison, but each has different omissions and tradeoffs.

Structuring the comparison around the same questions for every document keeps the comparison fair. Ask for the comparison in a consistent format, such as a table with one row per document and one column per question. This makes differences easier to spot than reading a paragraph of prose for each document:

Compare the three attached vendor documents for customer portal training.

Create one table with one row per vendor and these columns:

- Vendor
- Total price
- Earliest start date
- Delivery format and capacity
- Materials and follow-up support
- Conditions or limitations
- Missing information

Use short phrases so the table is easy to scan. Use only facts stated in the documents.

Do not combine details from different vendors. If a document does not answer a question, write "Not stated." Do not assume that something is included or excluded merely because it is not mentioned.

After the table, list up to three questions that should be answered before choosing a vendor. Do not recommend a vendor.

Total price	Earliest start date	Delivery format and capacity	Materials and follow-up support	Conditions or limitations	Missing information
US\$7,200; office-session travel extra, up to US\$600	August 24, 2026	Three 2-hour workshops; up to 45 people total; all online, or first at office and two online	Existing materials adapted to portal; one coaching call for managers	Final employee list due one week before first workshop; online sessions recordable; August 24 is after current launch target; details are from rough call notes	Payment dates; cancellation terms; later support; accessibility compliance; whether office session can be recorded
US\$9,600 fixed fee, including travel	August 12, 2026	Six 75-minute in-person workshops; up to 10 people per group; maximum 60 people total	Custom role-play exercises; accessible Word and PDF workbook; skills check; phone and email support for 60 days	Suitable room required; portal access due two weeks before first session; sessions not recorded; 10 business days' notice for cancellation or rescheduling; dates held only after signing; quote valid through August 5	Exact workshop dates; support hours and response times
US\$8,400 fixed fee	Week of August 17, 2026	Four live 90-minute virtual workshops; up to 50 people total; 9:00 a.m. to 5:00 p.m. Eastern Time	Workshop recordings; facilitator guide; two quick-reference sheets; email questions for 30 days	50% due when booking; balance after delivery; portal access due August 10; no evening or weekend sessions; five business days' notice to move a session without charge; August 17–21 hold only until August 3	Accessibility compliance; cancellation terms; support hours and response times

Figure 2.4F: Comparing unstructured documents into a structured table

Alongside the table, Claude flagged the following open questions to answer before choosing a vendor:

- On what dates would all workshops be completed?
- Do all materials and delivery methods meet the required accessibility standard?
- What are the complete cancellation, rescheduling, payment, and follow-up-support terms?

Retain links to supporting passages

A summary or extracted list is only useful if you can check where each item came from. Ask Claude to quote or reference the specific passage that supports each claim, action, or date it pulls out.



Watch out: a confident summary can still misstate a detail from the source. Confirm any figure, date, or decision that will affect someone's schedule or budget before you pass it along.



What to ask next: Flag any statement in the summary that you cannot trace back to a specific passage in the source material.

Try it now

Take a set of meeting notes or a long email thread. Ask Claude to extract every action, who owns it, and the deadline, as a short table. Then ask for a one-paragraph summary for someone who missed the meeting.



Learn more online: Use Claude for Outlook: <https://claude.com/docs/office-agents/outlook>. Install Claude for Outlook: <https://marketplace.microsoft.com/en-us/product/office/wa200010724>.

Check the result

- ☐ Does every extracted action have an owner and, where one was given, a date?
- ☐ Does the summary match the length and tone the reader needs?
- ☐ Can you trace each extracted item back to a specific passage in the source material?

Summarizing turns long text into a short, useful result. The next chapter does the same job for numbers. *Chapter 2.5* shows you how Claude can clean, calculate, and visualize a spreadsheet, so you can trust a chart before you present it.

Chapter 2.5 Analyze Spreadsheets and Data

Claude can run code in a secure analysis environment to clean, summarize, visualize, and make projections from structured data. This chapter follows Brightside Learning, a fictional training provider, as it decides which workshops to run more often next quarter. The walkthrough covers five steps: explain your data and question, clean the data, analyze and visualize results, validate the results, then decide:

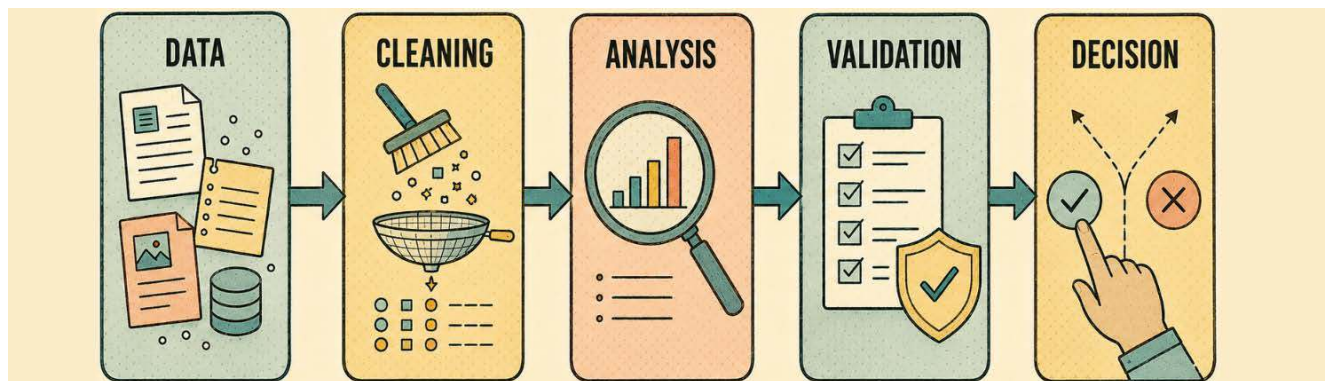


Figure 2.5A: Validation sits between analysis and decision on purpose. A chart is not a decision until you've checked the numbers behind it.

You need a spreadsheet or data file and a question you want that data to answer. Brightside Learning's workshop bookings give us both. By the end of this chapter, you'll be able to turn a raw spreadsheet into a checked analysis and chart, and you'll return to this same data in the final project chapter.



Sample files: If you don't have your own data to work with, download this document: <https://github.com/markjprice/claude-vb/tree/main/Files/Book2/2-5-sample-workshop-data.xlsx>. It contains a small, fictional sales spreadsheet of Brightside Learning's workshop sessions you can use to follow along. The sample workbook contains several sheets, but two are worth reviewing before you calculate anything. The Read Me sheet explains the workbook itself: the business question Brightside Learning wants answered, how each column is defined, and any notes about where the data came from. The Workshop Sessions sheet holds the raw data, with one row for each training session and columns for the course, delivery format, date, bookings, no-shows, fee per attendee, refunds, and satisfaction score. Ask Claude to read both sheets together so it understands the numbers and what they are meant to represent, not just one or the other.

Explain the data and the business question

Before you upload a spreadsheet, decide what question you want answered. “What does this spreadsheet show?” produces a vague response. “Which product line had the largest drop in sales last quarter?” gives Claude something specific to calculate.

Describe what each column means, especially if the headers are abbreviated or specific to your organization. A short explanation of the data saves several rounds of clarifying questions later:

Review the attached workbook, including the Read Me and Workshop Sessions sheets.

Before changing or calculating anything, explain:

- What one row represents
- What each column contains
- Which columns contain categories, dates, counts, money, ratings, or notes
- What the stated business question is
- How net revenue, attendance rate, and satisfaction could help answer it
- What the data cannot tell us

Use plain language for someone who does not work with data. List any missing definitions, assumptions, or questions that should be resolved before analysis.

Do not edit the workbook or calculate the results yet.

This prompt makes Claude establish the meaning and limits of the data before producing numbers:

📁 Workbook structure review ▾

What I found in the workbook

Read Me sheet explains the purpose and gives formulas; **Workshop Sessions** is the raw data, one sheet, 30 data rows (though one is a duplicate — more on that below).

What one row represents

Each row is one workshop session — a single class that Brightside Learning ran on a specific date, for a specific course, in a specific format (online or in person) and location. It is not one attendee and not one course overall; it's one scheduled event.

What each column contains

- **Session ID** — a unique code for each session (e.g., S-101), like a reference number
- **Session date** — when the session took place

Figure 2.5B: Claude explaining what rows represent and what columns contain

Clean and inspect the information

Real spreadsheets rarely arrive ready for analysis. Ask Claude to check for missing values, duplicate rows, and inconsistent formatting before running any calculation.

Ask to see what changed during cleaning. A cleaning step that drops rows or changes values without telling you can throw off every result that follows:

```
Inspect the Workshop Sessions sheet for data-quality problems.
```

```
Check for:
```

- Exact duplicate rows
- Missing values
- Inconsistent spelling, capitalization, or spacing
- Dates stored in different ways
- Values outside their expected range
- Bookings that exceed capacity
- Other values that may need human review

```
Keep the original sheet unchanged. Create:
```

1. A new sheet named Cleaned Sessions
2. A new sheet named Data Issues

```
In Cleaned Sessions:
```

- Remove only exact duplicate rows
- Standardize text values only when they clearly refer to the same course, format, or location
- Convert recognizable dates to real spreadsheet dates
- Do not fill missing values or change questionable figures by guessing

```
In Data Issues, record:
```

- Session ID
- Column
- Original value
- The problem found
- Any change made
- Whether human review is still needed

```
At the top of Data Issues, report the original row count, cleaned row count, number of duplicates removed, and number of unresolved issues.
```

```
Return the cleaned workbook as a downloadable Excel file.
```

Claude may ask permission before running commands or accessing files. Read what it plans to do before approving it. Use **Allow once** when you want to approve only that action. Grant continuing permission only when you understand what the permission covers and are comfortable allowing similar actions without another prompt:



Figure 2.5C: Choosing to allow once or always for Running commands



Watch out: Anthropic's Cowork safety guidance emphasizes reviewing permissions and matching oversight to the stakes of an action.

The source contains 29 rows representing 28 unique sessions. Claude should also find inconsistent labels, a text-formatted date, missing information, and a bookings figure above capacity:

Workbook structure review ▾

Here's the summary of what I found and did.

Original Workshop Sessions sheet: left completely unchanged (verified identical, row for row).

Cleaned Sessions (29 rows → 28 rows):

- Removed 1 exact duplicate row (S-114 appeared twice, identical in every column)
- Standardized spelling/capitalization that clearly referred to the same thing: "AI at work" → "AI at Work", "Excel essentials" → "Excel Essentials", trailing space trimmed from "Project Planning", "In person" → "In-person", "online" → "Online", "REMOTE" → "Remote"
- Converted S-120's text date ("June 12, 2026") into a real date value
- Did **not** touch missing values or the capacity/bookings mismatch — those need a human decision, not a guess

Data Issues log (12 entries, 3 s*** needing human review):

- S-107 — Refunds is blank (note says "Refund total not

Write a message...

+ Manual ▾ Sonnet 5 Medium ▾

Claude is AI and can make mistakes. Please double-check responses.

25 sample workshop data cleaned XLSX Google Drive

	A	B	C
1	Brightside Learning: workshop session data		
2			
4	Purpose of this sample		
5	This fictional workbook contains one row for each workshop session run from April through Ju		
6			
7			
8	Business question		
9	Which courses and delivery formats should Brightside Learning schedule more often n		
10			
11			
12	Measure	How to calculate it	Why it matters
13	Gross revenue	Bookings × fee per attendee	Shows sales before refunds
14	Net revenue	Gross revenue – refunds	Shows revenue kept after refu
15	Attendance rate	(Bookings – no-shows) ÷ bookings	Shows how many booked atte
16	Satisfaction	Average of stated scores from 1 to 5	Shows how attendees rated a
17			
18	All names, sessions, and figures are fictional. Currency values are in US dollars.		
19			
20			
21			

Read Me Workshop Sessions Cleaned Sessions Data Issues

Figure 2.5D: A cleaned spreadsheet artifact ready for analysis

Note the sheet tabs at the bottom of the artifact: **Read Me**, **Workshop Sessions**, **Cleaned Sessions**, and **Data Issues**. Review them as you read Claude's response to understand what it did.



Sample files: The spreadsheet with the cleaned sessions: <https://github.com/markjprice/claude-vb/tree/main/Files/Book2/2-5-sample-workshop-data-cleaned.xlsx>. The Cleaned Sessions sheet is the one you created in the previous step. It holds the same session data as the original Workshop Sessions sheet, but with exact duplicates removed, text values standardized, and dates converted to real spreadsheet dates. Nothing has been calculated yet: no revenue, no attendance rate, no summary. Use this sheet, not the original, so your calculations run on data that has already been checked rather than on raw, unverified rows.

Calculate and visualize useful results

Ask for the specific calculation you need, such as a total, an average, a trend, or a projection, rather than a general analysis. Ask for a chart only after you know which numbers matter, so the chart shows the comparison you actually need:

Using the Cleaned Sessions sheet, add these calculated columns with spreadsheet formulas:

- Attended: Bookings minus No-shows
- Gross revenue: Bookings multiplied by Fee per attendee
- Net revenue: Gross revenue minus Refunds
- Attendance rate: Attended divided by Bookings

If Refunds is missing, leave Net revenue blank and flag the session for review. Do not treat a missing refund value as zero.

Create a new sheet named Analysis containing:

1. A summary by course
2. A summary by delivery format

For each summary, calculate:

- Number of sessions
- Total bookings
- Total attended
- Overall attendance rate, calculated as total attended divided by total bookings
- Sessions with known net revenue
- Total known net revenue
- Average net revenue per session with known revenue
- Average satisfaction, excluding missing scores

Create a horizontal bar chart comparing average net revenue per session by course. Sort the courses from highest to lowest. State clearly that the chart uses only sessions with known refund information.

Below the summaries, write three concise observations that could help with next quarter's schedule. Separate facts from interpretations, and mention any missing or questionable data that limits the conclusions.

Keep all calculations visible in the workbook and return it as a downloadable Excel file.

I chose average net revenue per session for the chart because total revenue would unfairly favor any course that happened to run more often. The prompt also requests attendance and satisfaction, preventing the analysis from treating revenue as the only measure:

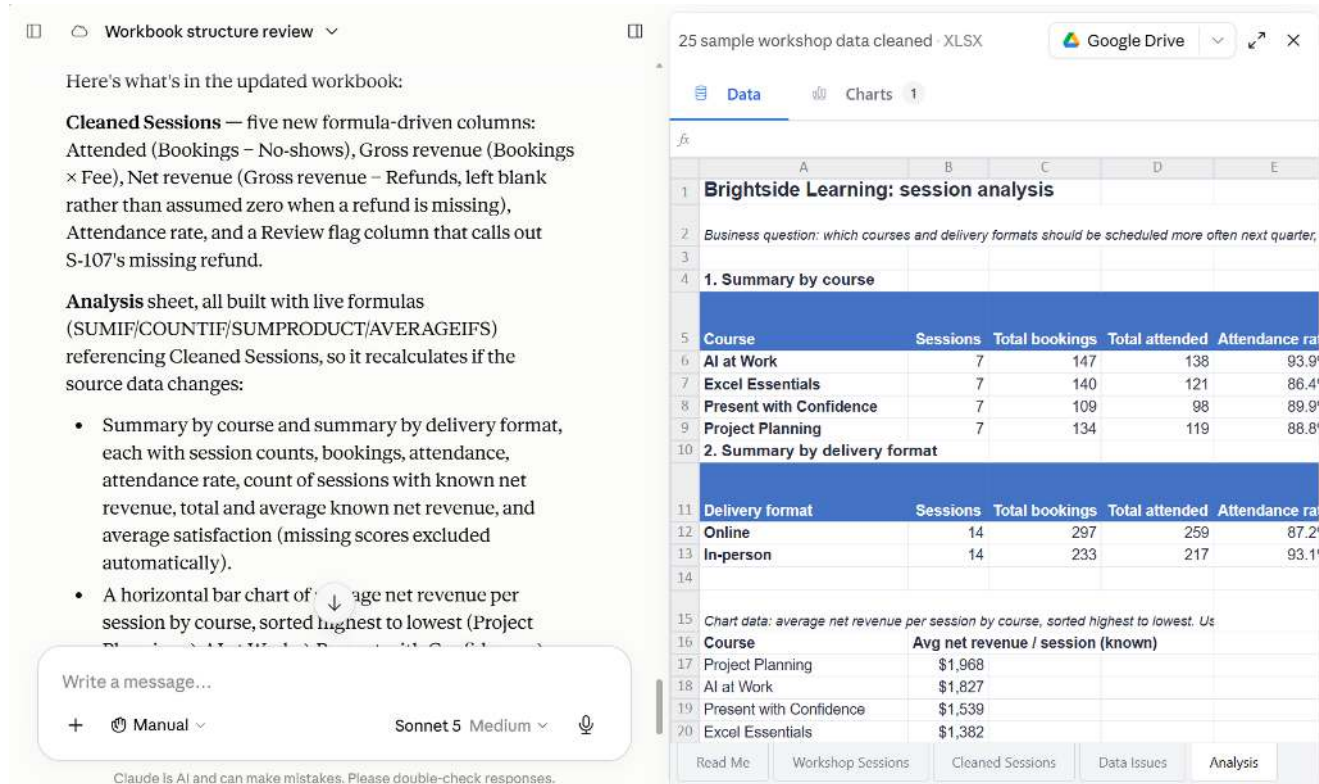


Figure 2.5E: Analysis sheet in the new Excel spreadsheet



Sample files: The final spreadsheet with the generated analysis: <https://github.com/markjprice/claude-vb/tree/main/Files/Book2/2-5-sample-workshop-data-analysis.xlsx>.

Compare the original columns with the generated analysis to confirm Claude used the data the way you intended. Select **Charts (1)** to see the chart it generated.



Learn more online: *Chapter 2.3* introduces the Claude add-ins for Word, Excel, PowerPoint, and Outlook. Use Claude for Excel: <https://claude.com/docs/office-agents/excel>.

Check calculations and interpretations

Spot-check a calculation against the raw data before you use it in a decision. Pick a few rows, do the arithmetic yourself, and compare the result.

An average or trend can look reasonable while resting on a small sample, an outlier, or a misread column. Ask Claude to state how many rows a calculation is based on and whether any values were excluded.



Good practice: Ask Claude to explain its method in plain language before you trust the result. If you can't follow the explanation, ask for it again in simpler terms rather than accepting a number you don't understand.



Real-world example: A shop manager uploads a year of sales data and asks Claude which product line lost the most ground last quarter, then checks the answer against the three lowest-performing rows by hand before adjusting orders.

Try it now

Upload a spreadsheet you already have and describe one specific question you want it to answer. Ask Claude to clean the data, show you what changed, then calculate and chart the result.



What to ask next: Which single number in this analysis would change my decision if it turned out to be wrong?

Check the result

- ☐ Do the totals or averages match a rough estimate you can calculate by hand?
- ☐ Did the cleaning step change or remove any rows you needed to keep?
- ☐ Does the chart compare the values you actually asked about?

With a checked analysis in hand, it's time to switch from numbers to a different kind of visual. *Chapter 2.6* covers generating and editing images, along with the permission and copyright questions you need to answer before you use one.

Chapter 2.6 Create Visual and Interactive Content

With a checked analysis in hand from *Chapter 2.5*, it's time to move from numbers to pictures. This chapter shows you two different ways to add a visual to your work: build a diagram or an interactive visual with Claude itself, or use Claude to write a strong prompt for a separate image generator when you need a photo or an illustration:

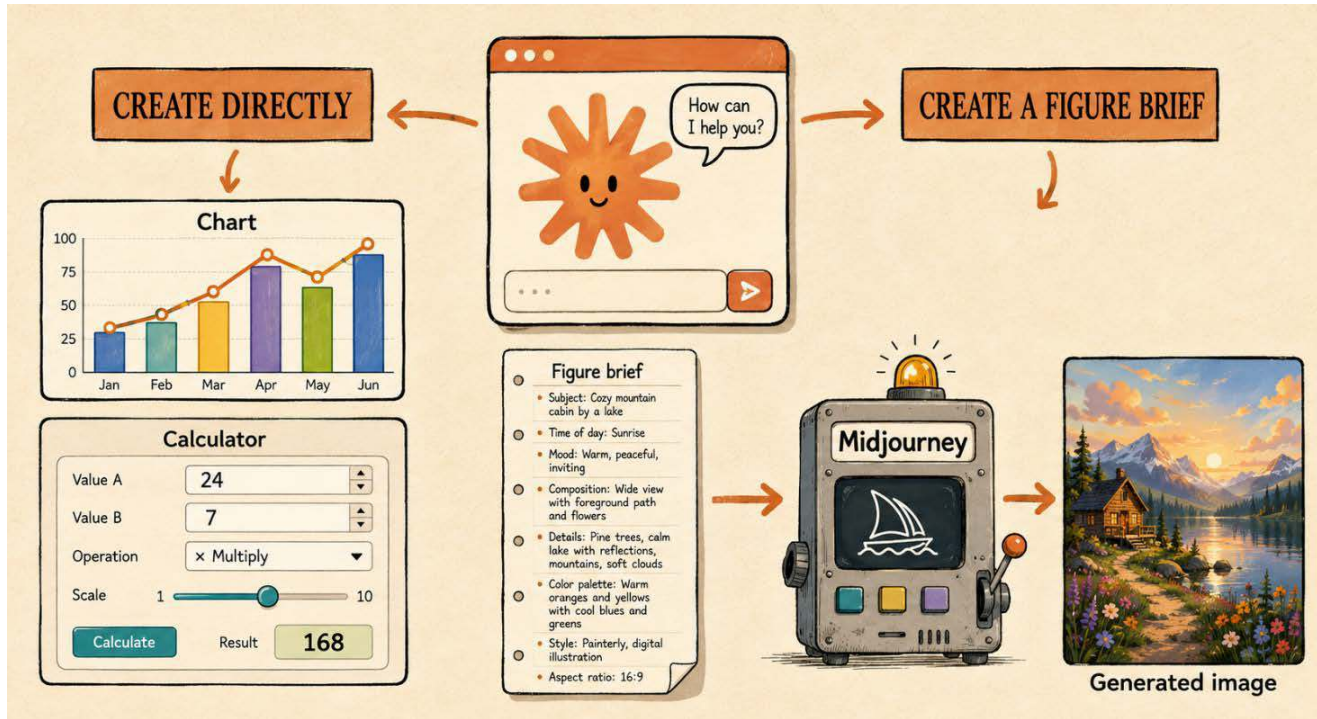


Figure 2.6A: Two ways Claude can help you create visual and interactive content

By the end of this chapter, you'll know which visual jobs Claude can do inside your conversation, why Claude won't produce a photo or illustration the way some other AI assistants like ChatGPT do, and how to get a useful result from an outside image generator when you need one.

You don't need any new software to build a diagram with Claude. If you want to try an outside image generator, you'll need a separate account with that service. Most offer a free trial or a limited number of free images before they ask you to pay.



Watch out: Check your employer's policy and the image generator's license before you use an AI-generated image in published or commercial work. Many generators restrict commercial use on their free plans, and disputes over how some generators were trained are still working their way through the courts.

See what Claude can build directly in the chat

Claude can create diagrams, charts, and other interactive visuals right inside a conversation. When Claude builds one of these, it opens next to the chat in a separate panel. You can view the artifact, ask Claude to change it, and see the update right away, without starting a new conversation.

An artifact is built from code, usually HTML, SVG, or a charting library, not from a photo. That makes it a good match for an organization chart, a flowchart of a process, a timeline, a simple interactive calculator, or a chart built from data you already have. It's not a way to produce a photorealistic picture or an illustration.



Real-world example: A project manager asks Claude to turn a list of project milestones into an interactive timeline. Claude builds the timeline as an artifact. The project manager then asks for the milestones to be grouped by department and labeled clearly, since the printed version of her report won't use color, before she shares it with her team.

You don't need any design skill or coding knowledge to get a useful artifact. Describe what you want in plain language, and Claude builds it. Three simple visuals are a good place to start.

Map a process as a flowchart

If you can list the steps in a process, in order, Claude can turn that list into boxes and arrows you can follow at a glance. This works well for anything with a clear sequence and a few decision points, such as how your team approves a purchase order or how a support ticket moves from submitted to resolved:

Turn these steps into a flowchart: request submitted, manager reviews, finance checks the budget, purchase order issued. Add a branch for what happens if finance rejects the request.

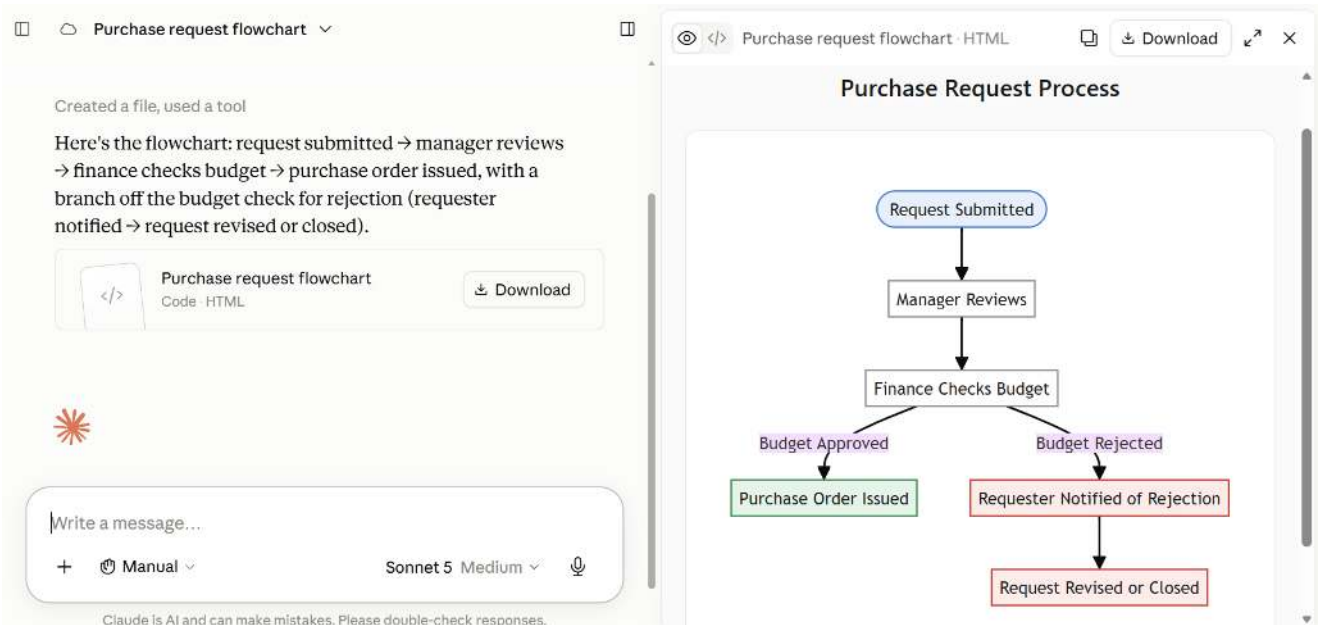


Figure 2.6B: A flowchart built using a Mermaid diagram embedded in a web page

Build a timeline for a project or an event

List the milestones and their dates, and ask Claude to lay them out along a line. A timeline is often clearer than a table when you want your audience to see how much time sits between two milestones, not just the dates themselves:

Build a timeline for a product launch: planning finished March 1, prototype ready April 15, beta testing May 1 to May 20, launch June 1.

Create a small interactive calculator

An artifact isn't limited to a static picture. Claude can build a small tool that takes a number you type in and shows a result right away, such as a tip calculator, a simple way to split a shared expense, or a currency converter using a rate you supply:

Build a calculator that splits a restaurant bill between a group of people, with a field for the tip percentage.

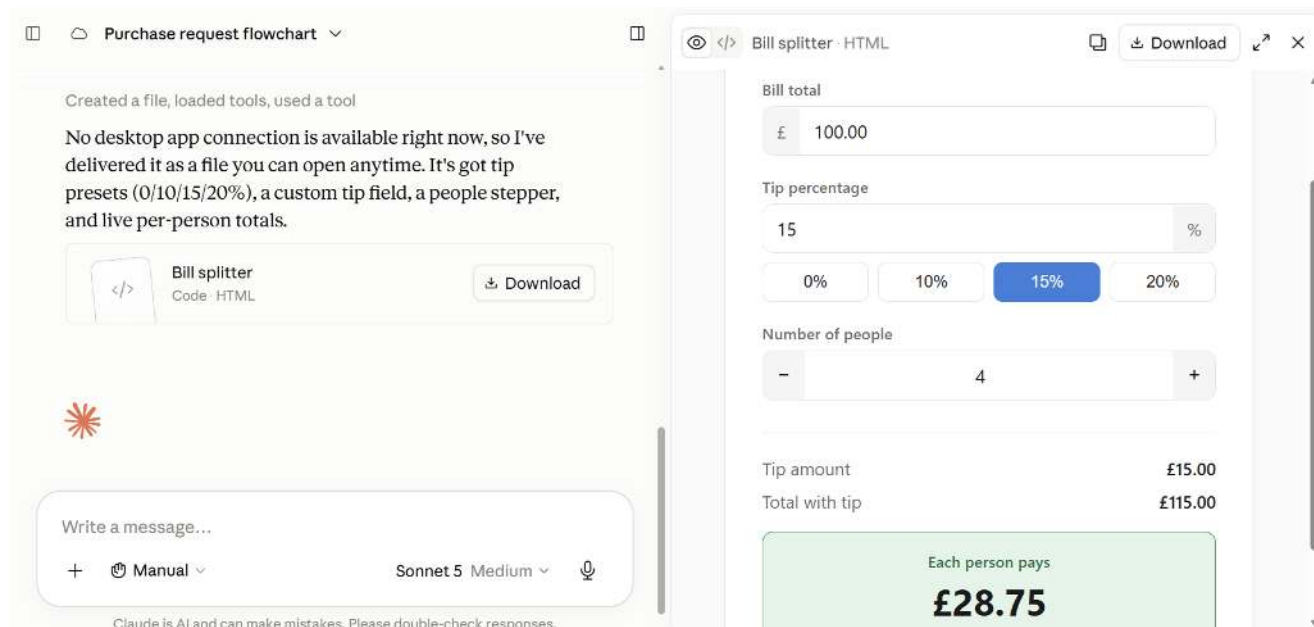


Figure 2.6C: An interactive calculator for splitting a restaurant bill



Good practice: Describe the purpose of the visual, not its exact layout. Tell Claude what it needs to show and who will read it, then let Claude propose a design. You can always ask for a different layout once you see the first version.

Know why Claude doesn't generate native photorealistic and illustrations

Some AI assistants, like ChatGPT, have an image generator built into the chat. You type a description and get back a photo-style picture or an illustration. Claude works differently. Claude can read and describe an image you upload, and it can write code that draws a diagram or a chart, but it has no connected tool that turns a text description into a photo or an illustration.



Real-world example: OpenAI publishes worldwide ChatGPT usage data through a project called OpenAI Signals. Its August 6, 2026 report named images and other media as ChatGPT's fastest-growing use case. By the second quarter of 2026, 7.8% of all ChatGPT messages worldwide asked it to create, edit, or find media. <https://openai.com/index/how-the-world-is-putting-chatgpt-to-work/>.

Anthropic has focused Claude’s development on reasoning, writing, and coding tasks rather than image generation. Because of that choice, if you ask Claude to “generate a picture of a mountain at sunset,” it will tell you it can’t and suggest another route: building a diagram if a diagram would do the job, or recommending an image generator if you need a photo or an illustration.



Learn more online: Anthropic’s help center article *Can Claude produce images?* confirms what Claude can and can’t create, and points to workaround suggestions: <https://support.claude.com/en/articles/9002504-can-claude-produce-images>.

Choose an image generator for photos and illustrations

When you need a photo-style image or an illustration, you’ll need a separate tool. A few well-known options include:

- Midjourney, known for painterly and highly stylized images, used through its own web app: <https://www.midjourney.com/>. Midjourney does not offer a free plan.
- ChatGPT’s built-in image tool, from OpenAI, which generates a picture directly inside a ChatGPT conversation: <https://chatgpt.com/>.
- Google’s Gemini app, which includes Google’s image generator Nano Banana and can also edit a picture you upload: <https://gemini.google.com/>.
- Adobe Firefly, built into Photoshop and other Adobe apps, and trained to be a safer choice for many commercial uses: <https://firefly.adobe.com/>. One benefit of Adobe models like Firefly Image 5 is that they are commercially safer to use, meaning they are trained on a dataset of licensed content, such as Adobe Stock and public domain content where copyright has expired. Adobe says, “We do not mine the web or video hosting sites for content. We only train on content where we have rights or permission to do so.”
- Ideogram, from Davinci AI, known for rendering readable text inside an image, such as a poster or a logo: <https://davinci.ai/image-generator/ideogram-4>.

Each generator has its own style, price, and license. Check the generator’s current terms before you use an image in client work or anything you plan to sell.



Good practice: Before you subscribe to an image generator, run the same prompt through two or three free trials. Compare the results for the specific style you need. One generator’s strength for a realistic portrait can be another generator’s weakness.

Use Claude to write a strong prompt for another generator

Even though Claude can’t produce the image itself, it can help you write the prompt. Image generators respond best to a description that states the subject, the setting, the style, the lighting, and the framing. Claude is good at turning a rough idea into a prompt with that level of detail.

Tell Claude which generator you plan to use. Some tools respond well to a plain description, while others expect a list of keywords and camera terms. A short back-and-forth usually works better than one long, overloaded prompt:

```
I need an image for a blog post about remote work. I want something warm and welcoming, not the usual stock photo of someone on a video call. I will use Midjourney to create the image.
```

Claude can offer several prompt options, such as a home office with morning light and a half-finished cup of coffee, or a small kitchen table doubling as a desk with a view of a garden:

Here are a few Midjourney prompt options that avoid the video-call cliché and lean into warmth:

Option 1 — Cozy home setup

```
A sun-drenched home office corner, warm morning light streaming through a window onto a wooden desk, a steaming mug of coffee, an open notebook, a small potted plant, soft golden tones, lived-in and cozy, shallow depth of field, editorial photography style --ar 16:9 --v 6
```

Option 2 — Human, not tech-focused

```
Hands writing in a notebook at a kitchen table bathed in warm afternoon light, a laptop closed nearby, soft bokeh background, cozy domestic atmosphere, warm color palette, natural light photography --ar 16:9 --v 6
```

Figure 2.6D: Claude offering prompts that are optimized for Midjourney for image generation.

You continue the conversation:

```
The kitchen table version is close. Change it to evening light, and drop the garden view.
```

Claude rewrites the prompt with the evening light and the garden view removed, ready to paste into your chosen generator:

```
Hands writing in a notebook at a kitchen table bathed in warm evening light, a laptop closed nearby, soft bokeh background, cozy domestic atmosphere, warm color palette, natural light photography --ar 16:9 --v 6
```

I do not have a Midjourney account, but I do have a Google AI Pro account, so I told Claude that I would use Google Nano Banana and I used Google AI Studio and its Nano Banana 2 Lite model:

```
I need an image for a blog post about remote work. I want something warm and welcoming, not the usual stock photo of someone on a video call. I will use my Gemini account and Nano Banana to create the image.
```

Claude responded with four suggested prompts:

```
Here are prompt options tailored for Gemini/Nano Banana – it tends to respond well to natural, descriptive sentences rather than parameter flags, so I've written these more conversationally than the Midjourney versions.
```

```
...
```

I picked one and *Figure 2.6E* shows the result in Google AI Studio:

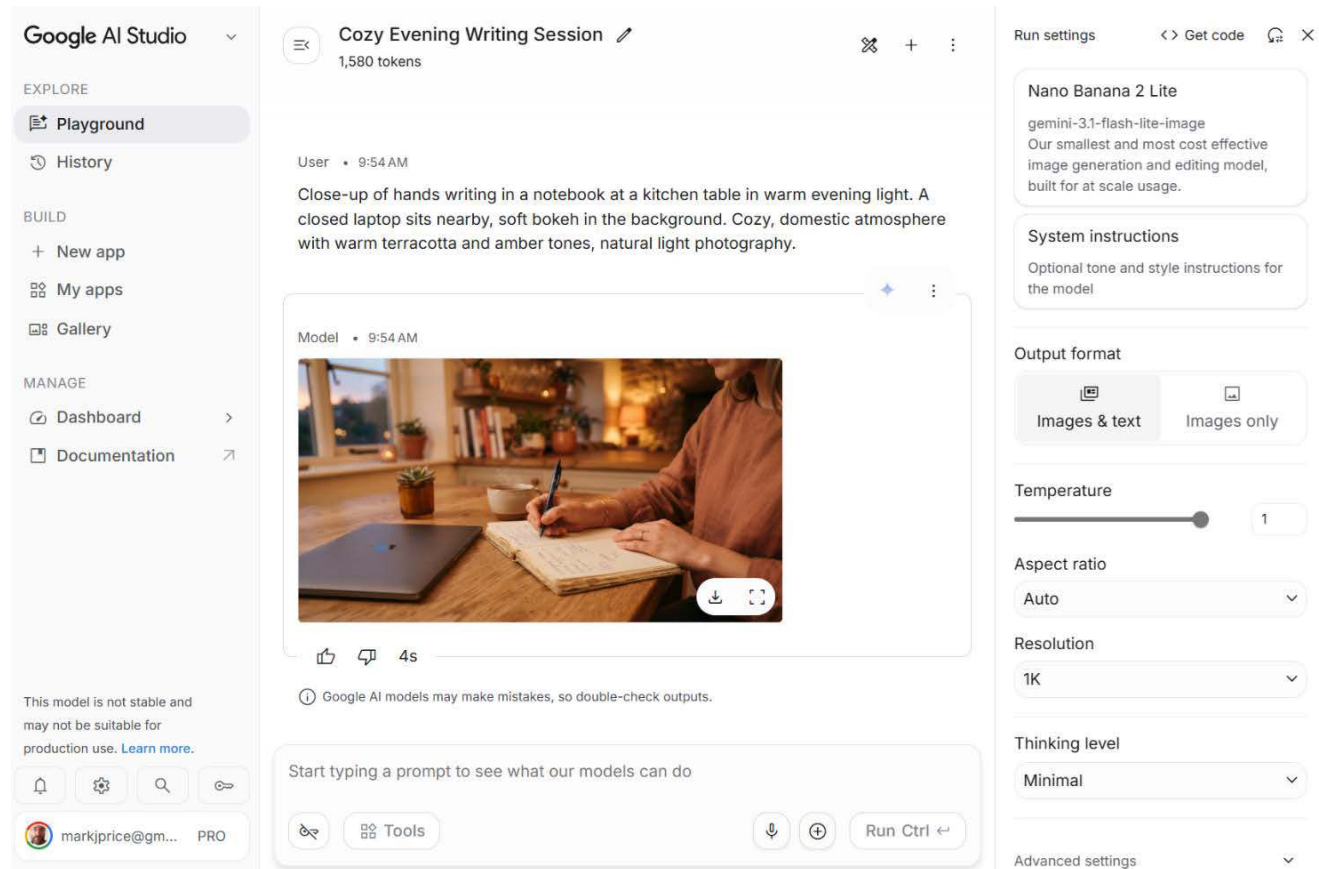


Figure 2.6E: Nano Banana 2 Lite generating an image based on a Claude prompt



What to ask next: Write three different prompts for this idea, each in a different visual style.

Bring the finished image back into Claude

Once you have an image, upload it to Claude to check it and put it to work. Claude can describe the image back to you and tell you whether it matches your brief. Claude can also help you write alt text for the image or place it into a document or presentation you're already drafting.

Save the finished image in a folder you can find again before you move on. You'll want it close by when you build the presentation in *Chapter 2.7*.

Try it now

Pick one topic and build two visuals for it: an interactive visual you make directly in Claude, and a photo or illustration you get from an outside image generator with Claude's help writing the prompt.

If you don't have your own idea, continue the Brightside Learning example from *Chapter 2.5*. Ask Claude to turn the workshop attendance numbers into an interactive chart. Then ask Claude to write a prompt for an illustration to open the year-end report, showing a diverse group of adult learners in a bright classroom.



What to ask next: Give me two more style options for this illustration prompt: one more formal, and one more playful.

Check the result

- ☐ Does the interactive visual update correctly when you ask Claude for a change?
- ☐ Does the image generator's picture match the subject, style, and mood you described?
- ☐ Did you check the generator's license before saving the image for later use?

With a chart Claude built in a spreadsheet and a picture from an outside generator, you have the raw material for a stronger document. *Chapter 2.7* brings together the writing and visual skills from the last two chapters to help you build a slide deck that leads your audience to a decision.

Chapter 2.7 Create Presentations and Visual Reports

A presentation should lead an audience to a decision, not just display information. This chapter covers defining the audience and desired decision, building a narrative, generating the slides and notes, and inspecting the finished slide deck before you deliver it.

Figure 2.7A traces this path from evidence through argument to a slide deck aimed at one audience decision.

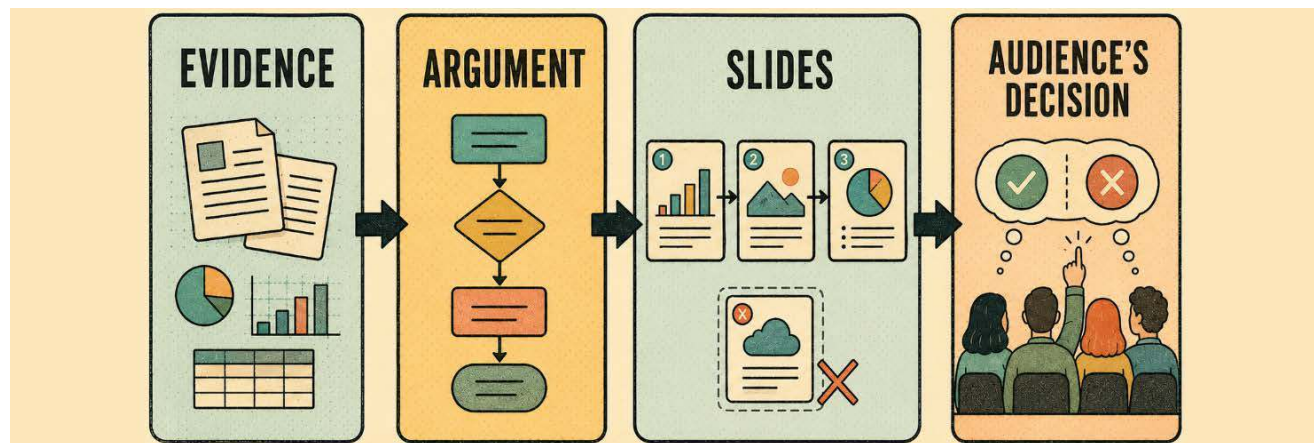


Figure 2.7A: Each slide should support one step in this chain. A slide that doesn't connect to the argument is a candidate for removal.



Real-world example: A department head builds a budget request deck by first asking Claude for a narrative outline built around one decision, whether to approve a new hire, and only then asking for slides.

You need a topic, an audience, and a rough sense of what you want that audience to decide or do. By the end of this chapter, you'll be able to turn a brief into a reviewed slide deck with speaker notes.

If you don't have your own idea, I recommend one scenario that continues the Brightside Learning example from *Chapter 2.5*.

This scenario shows the difference between analyzing data and communicating a recommendation. The spreadsheet may contain dozens of results, but the audience needs only the evidence that helps them make the scheduling decision.

Define the audience and desired decision

Before asking for slides, decide who will watch the presentation and what you want them to approve, choose, or understand afterward. A presentation for a client deciding whether to sign a contract needs a different structure than one for a team reviewing progress.

State the audience and the decision in your first message. This shapes everything that follows, including how much background the deck needs and how the closing slide should read.

Include what the audience already knows, how much time they have, and what matters to them. A senior manager may need the recommendation, supporting evidence, risks, and requested decision. They probably do not need a detailed explanation of every calculation.

You can turn the spreadsheet analysis into a presentation for a real decision:

- **Topic:** Decide which Brightside Learning workshops to offer more often next quarter.
- **Source material:** The cleaned and analyzed version of `2-5-sample-workshop-data.xlsx`.
- **Audience:** Brightside Learning's managing director and program manager. They understand the business but have not studied the spreadsheet.
- **Desired decision:** Approve the recommended scheduling priorities for next quarter, including which courses or delivery formats to increase, maintain, or reconsider.
- **Evidence:** Known net revenue, attendance rate, satisfaction, and data-quality limitations.
- **Format:** A seven-slide presentation designed for a seven-minute meeting.

Use the following prompt with the analyzed workbook from *Chapter 2.5*:

I need to prepare a short presentation based on the attached Brightside Learning workbook.

Topic: Which courses and delivery formats should Brightside Learning schedule more often next quarter?

Audience: The managing director and program manager. They understand the workshop business but have not reviewed this analysis.

Desired decision: Approve the recommended scheduling priorities, including which courses or formats to increase, maintain, or reconsider.

Meeting: I will have seven minutes to present, followed by questions.

Before creating any slides, prepare a presentation brief that states:

- What the audience needs to understand
- The decision I am asking them to make
- The evidence that will matter most
- Questions or objections the audience may raise
- The information that can be left out
- The appropriate tone and level of detail

Use only information found in the workbook. Identify any missing, questionable, or incomplete data that may limit the recommendation. Do not create the slides yet.

Review the brief before continuing. Check that the requested decision is narrow enough for one meeting. “Discuss next quarter” is vague. “Approve the recommended scheduling priorities” gives the presentation a clear destination:

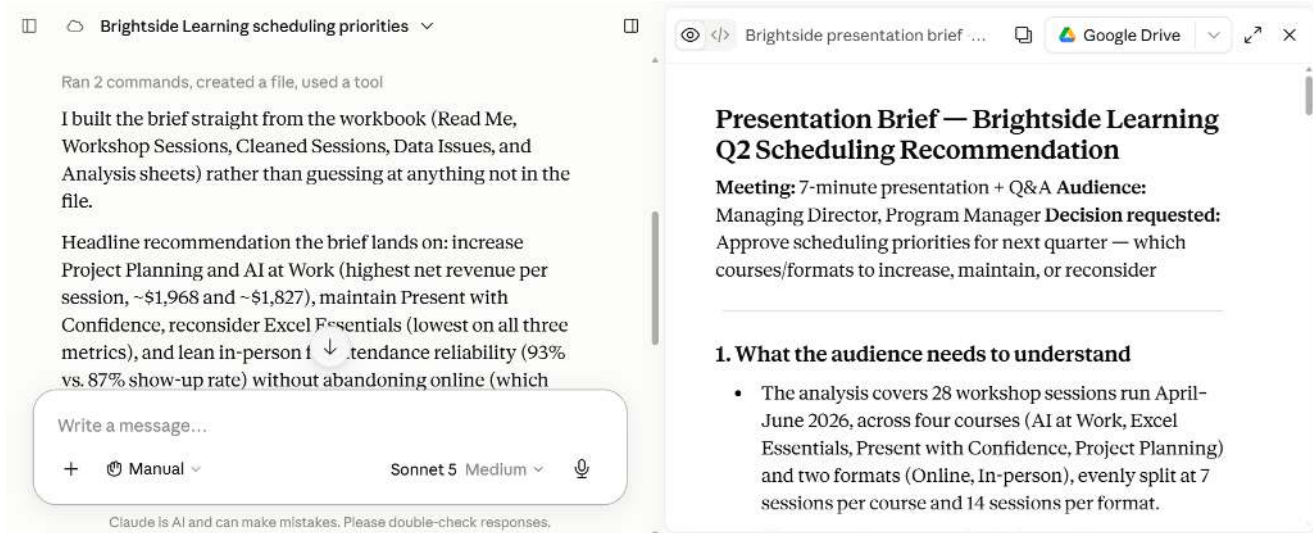


Figure 2.7B: A presentation brief based on the analyzed spreadsheet

Build the story before creating slides

A strong presentation follows a line of reasoning: the problem or opportunity, the evidence, the options, and the recommendation. Each slide should prepare the audience for the next one. If Claude creates the complete deck immediately, it may repeat information, introduce the recommendation too late, or fill slides with facts that do not support the decision.

Ask Claude for an outline of this narrative before generating any slides, the same way you would ask for a document outline first:

Using the presentation brief and the attached workbook, create an outline for a seven-slide presentation.

The presentation should lead the audience from the business question to a clear scheduling decision.

For each slide, provide:

- Slide number
- Short slide title
- The one point the audience should remember
- Evidence from the workbook that belongs on the slide
- A suitable visual, such as a chart, table, or simple diagram
- The purpose of the speaker notes

Include:

1. A title slide that states the decision needed
2. The business question and measures used

3. Performance by course
4. Performance by delivery format
5. Recommended scheduling priorities
6. Data limitations and risks
7. The decision and next steps

Use only facts supported by the workbook. Do not invent a number of future sessions, a budget, a cause, or a forecast.

Keep the outline concise. Do not create the finished presentation yet.

A useful slide title states the slide's message. For example, "Live online sessions have the strongest attendance rate" tells the audience more than "Delivery format analysis." The exact title should, of course, depend on the workbook results.

Check that each proposed chart answers a question. A decorative chart that does not help the audience decide can be removed:

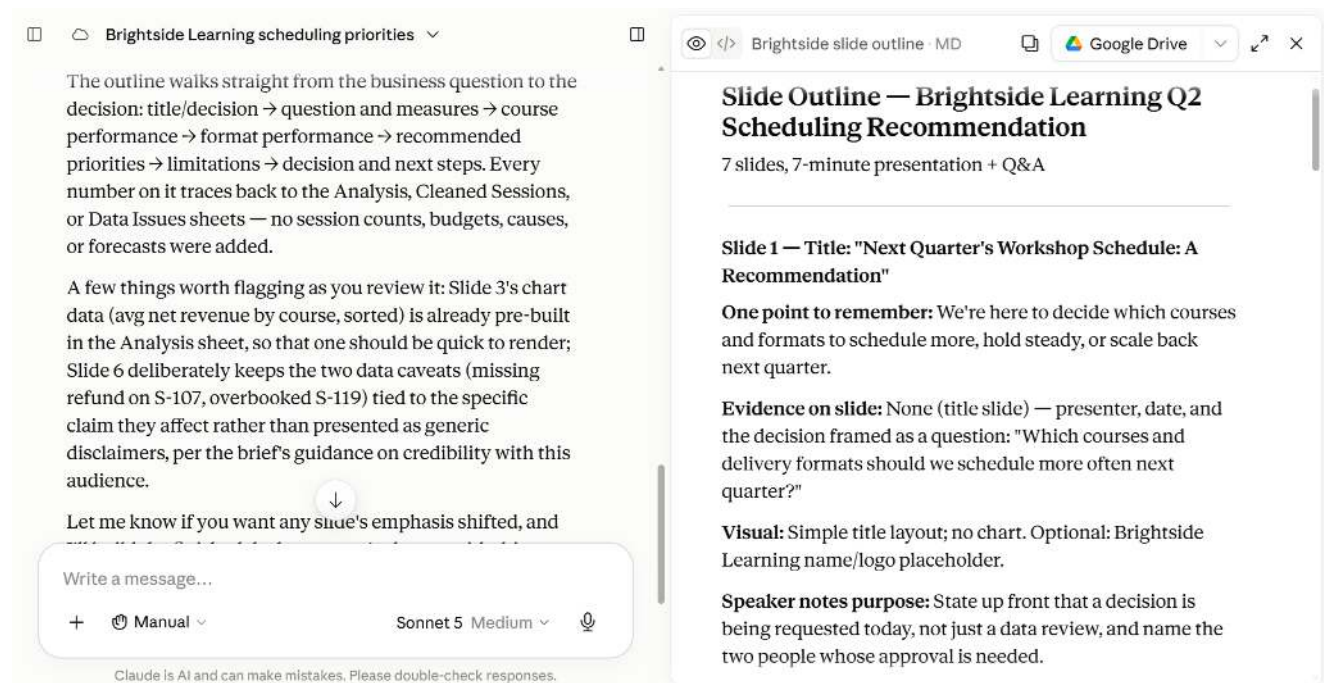


Figure 2.7C: A slide outline for a presentation with a scheduling recommendation

Generate slides, notes, and supporting visuals

After approving the outline, ask Claude to build the presentation. State how much text each slide may contain and which visual formats are suitable. Charts should represent the source data accurately. Generated artwork may support a theme or illustrate a broad idea, but it should not be used to represent exact figures. Ask for notes that expand on what you'll say, not a script you'll read word for word:

Using the attached workbook, the agreed presentation brief, and the approved seven-slide outline, create the complete presentation as an editable PowerPoint file.

Design requirements:

- Use a clean 16:9 layout
- Use large text that will be readable in a meeting room

- Give each slide one clear message
- Keep body text to no more than five short bullets
- Use short, message-based slide titles
- Use a consistent layout, type style, and color palette
- Do not rely on color alone to communicate meaning
- Do not add decorative images that compete with the evidence

Charts and supporting visuals:

- Create charts from the workbook data
- Label every chart clearly
- Show whether revenue figures include only sessions with known refund information
- Use overall attendance rate rather than an unweighted average of session percentages
- Exclude missing satisfaction scores from averages
- Keep charts and diagram elements editable where possible
- Use a small table or decision matrix to show whether each course or format should be increased, maintained, or reconsidered
- Do not use a generated image to represent numeric results

Speaker notes:

- Add brief speaker notes to every content slide
- Explain the slide's main point
- State the evidence that supports it
- Mention any limitation the presenter should disclose
- Suggest a natural transition to the next slide

Accuracy:

- Use only facts from the workbook
- Do not invent causes, forecasts, budgets, or future results
- Mark an unsupported recommendation with [REVIEW]
- Preserve uncertainty where source data is missing or questionable

End with a slide that states the requested decision and the next action.

Return the finished presentation as a downloadable PowerPoint file.

The speaker notes should add useful context rather than repeat the slide word for word. They can remind the presenter about a missing value, explain how a measure was calculated, or prepare for a likely question.

Supporting visuals may include a horizontal bar chart comparing courses, a second chart comparing delivery formats, a small increase, maintain, or reconsider decision matrix, and a simple next-steps timeline. The charts should do most of the work. A presentation based on business data does not need an AI-generated photograph on every slide.

Figure 2.7D shows the generated slide deck in the artifacts panel:

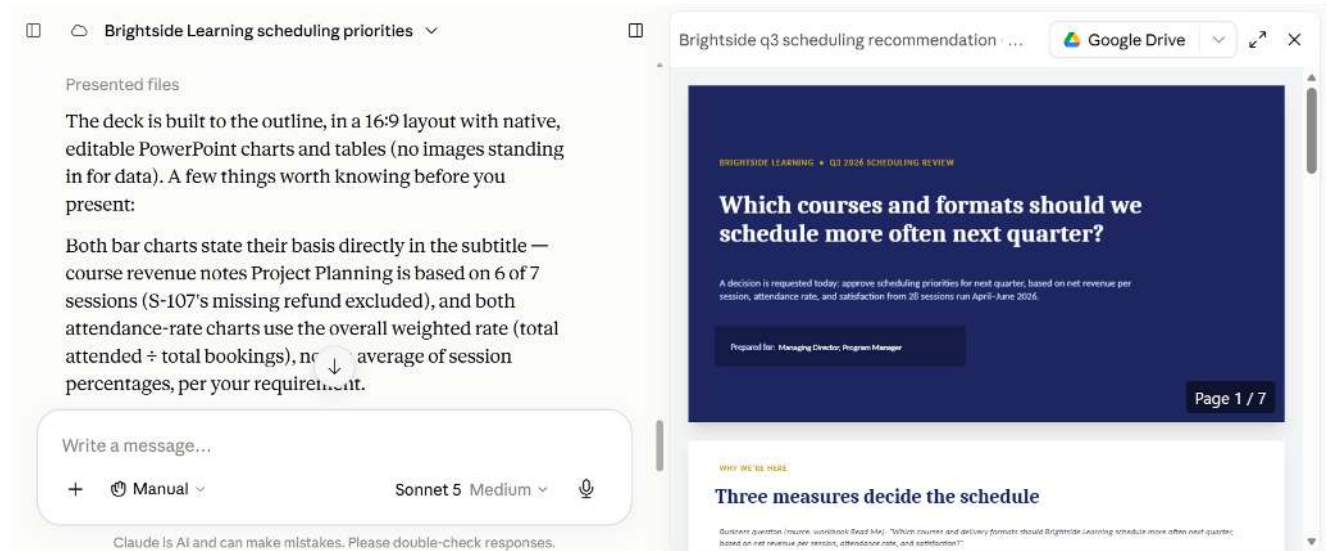


Figure 2.7D: Compare the outline with the finished slides to confirm every planned point made it into the deck



Learn more online: Use Claude for PowerPoint: <https://claude.com/docs/office-agents/powerpoint>.

Inspect the complete presentation before delivery

A presentation can look finished while still containing incorrect figures, weak conclusions, or words that are too small to read. Review the complete deck against the original source, not only against Claude's summary of that source.

Read every slide in order, not just the ones you asked for changes on. Check that numbers match across slides, that charts show what the text claims, and that the opening and closing slides connect.

The same material can become a one-page visual report for people who will not attend the meeting. Ask Claude to preserve the recommendation, evidence, limitations, and next action, but redesign them for reading rather than presenting. A report must make sense without speaker notes.



Good practice: Ask a colleague unfamiliar with the material to look at the deck and tell you what decision they think you're asking for. If their answer doesn't match your intended decision, revise the narrative before the real presentation.

Try it now

Pick a real presentation you need to give. State the audience and the decision you want from them, ask for a narrative outline, review it, then generate the full deck with speaker notes.



What to ask next: Which slide in this deck could I remove without weakening the argument?

Check the result

- ☐ Does each slide support one step in the argument, rather than repeating information from another slide?
- ☐ Do the speaker notes help you present, rather than reading like a script?
- ☐ Would the intended audience reach the decision you were aiming for after seeing this deck?

So far, you've worked with material you already had in hand. *Chapter 2.8* opens the door to information Claude can reach on its own, once you or your administrator connect a source such as email or a shared drive.

Chapter 2.8 Add Plugins, Connectors, and Skills

Claude can do more than respond using its general knowledge and the files you upload. You can add capabilities that help it follow specialized workflows, work with external services, and create particular types of output.

These capabilities are organized around three related ideas:

- A **plugin** is a package you can discover and install
- A **connector** connects Claude to information or actions in another service
- A **skill** gives Claude reusable instructions for completing a particular kind of work

You do not need to understand how these capabilities are built before using them. However, you should understand what you are installing, what information it can access, and whether it can take actions outside Claude.

Understand how the pieces fit together

Think of a plugin as a box of capabilities prepared for a particular kind of work. Depending on its purpose, the plugin may contain one or more skills, one or more connectors, one or more sub-agents, or a combination of them:

Capability	What it provides	Simple example
Plugin	An installable package containing capabilities for a workflow	A presentation plugin containing instructions and supporting tools
Connector	A connection to an external service, its information, or its actions	Google Drive, SharePoint, Slack, or GitHub
Skill	Reusable instructions, examples, and resources for a particular task	A skill for creating and checking a presentation
Sub-agent	A separate Claude instance that your main session hands a specific task off to, and then reports a result back	It handles noisy, large-scale software development work on its own and only reports back a summary, keeping the main conversation clean

Table 2.8A: Plugin capabilities and what they provide

A skill is like a written playbook. It might tell Claude how to inspect a spreadsheet, calculate results, create suitable charts, and check the finished workbook. The skill does not necessarily connect to another company's service.

A connector is more like a doorway. It may allow Claude to find a file in Google Drive, read a page in Notion, search messages in Slack, or create something in another service. The connector can only use the access you and your organization have allowed.

A plugin brings these pieces together so that you can install a useful capability without assembling every part yourself.

Figure 2.8A traces the path from your request, through Claude and the connector, to the specific information it selects and the output it produces:

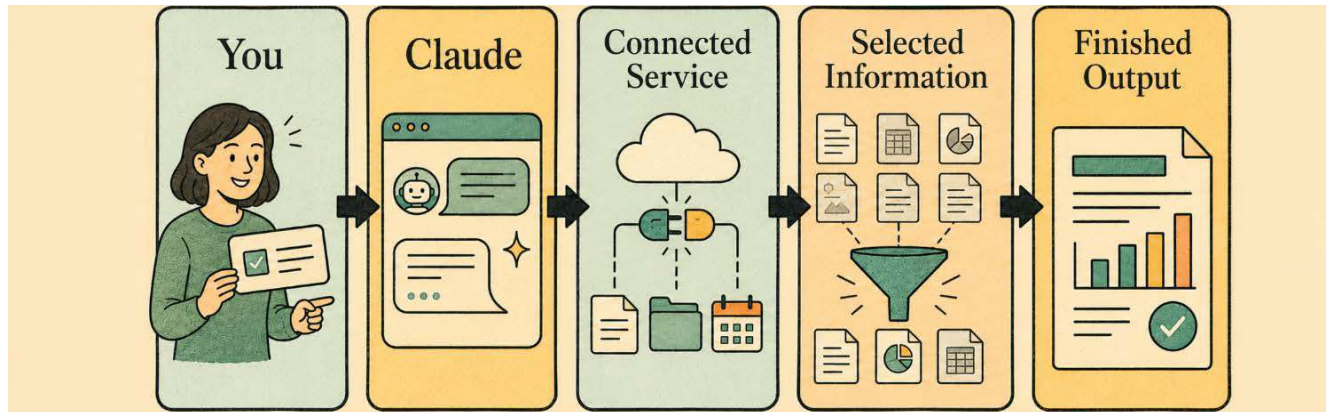


Figure 2.8A: Every connector passes through this same chain. Reviewing the selected information before the output stage catches most problems early.

You need access to at least one approved plugin, or an administrator who can approve one. By the end of this chapter, you'll be able to pull relevant workplace information into a task while keeping control over what Claude sees.

Customize Claude

Customize is the main place to discover capabilities for Claude.

Open **Customize** from the Claude sidebar or settings. The exact position may vary between the web, mobile, and desktop versions:

1. In the sidebar, select **Customize**, and note there are four sections: **Skills**, **Connectors**, and **Plugins**.
2. Select **Plugins** and note some plugins with skills are already installed:

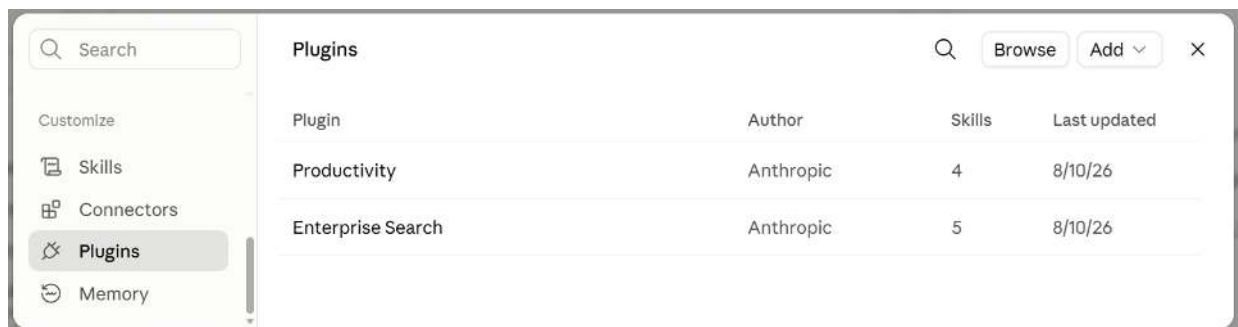


Figure 2.8B: Preinstalled plugins with skills

3. Select the **Productivity** plugin to view its details:

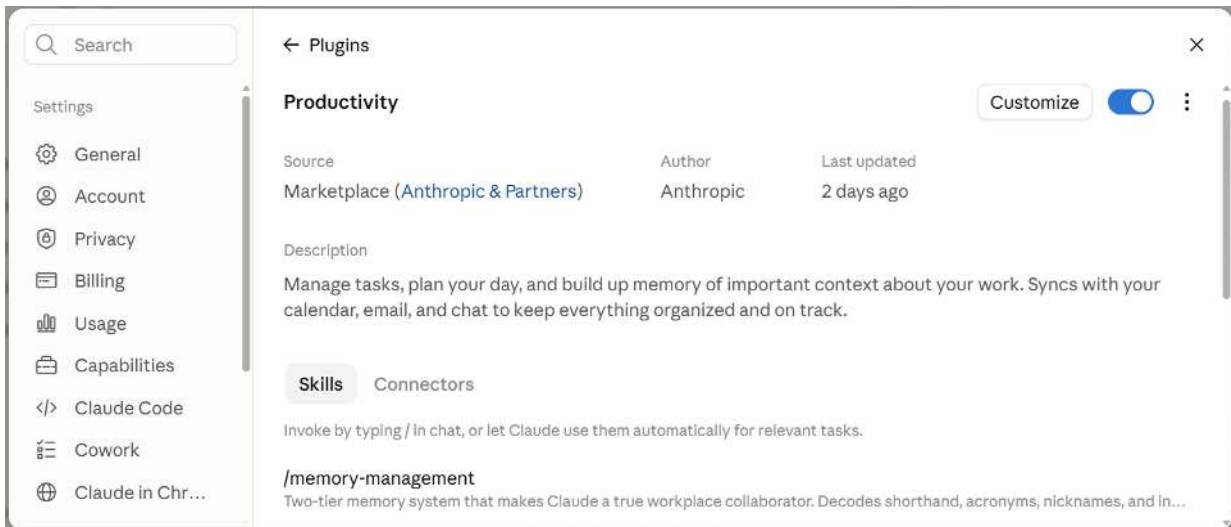


Figure 2.8C: Productivity plugin details including skills

4. Select **Connectors**, and note there are buttons to install and configure the connectors you want to use while working with Claude:

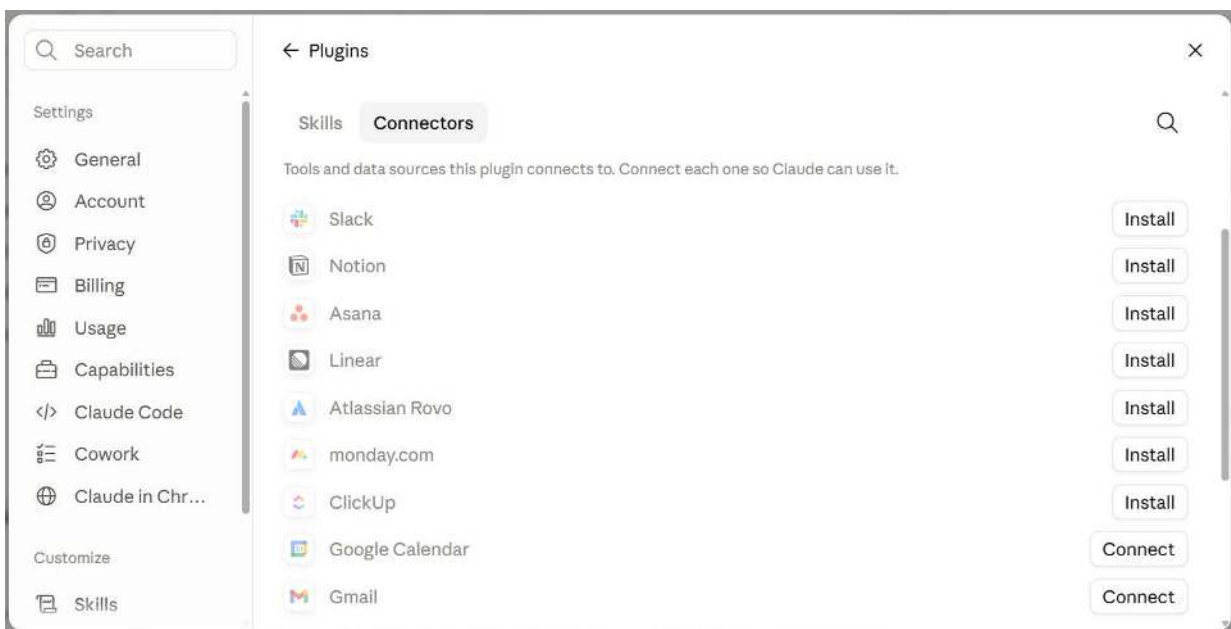


Figure 2.8D: Connectors for external systems like Slack and Gmail

5. Select the ← **Plugins** button to go back.
6. Select the **Browse** button.

7. In the **Directory**, note there are many plugins from Anthropic and others to add:

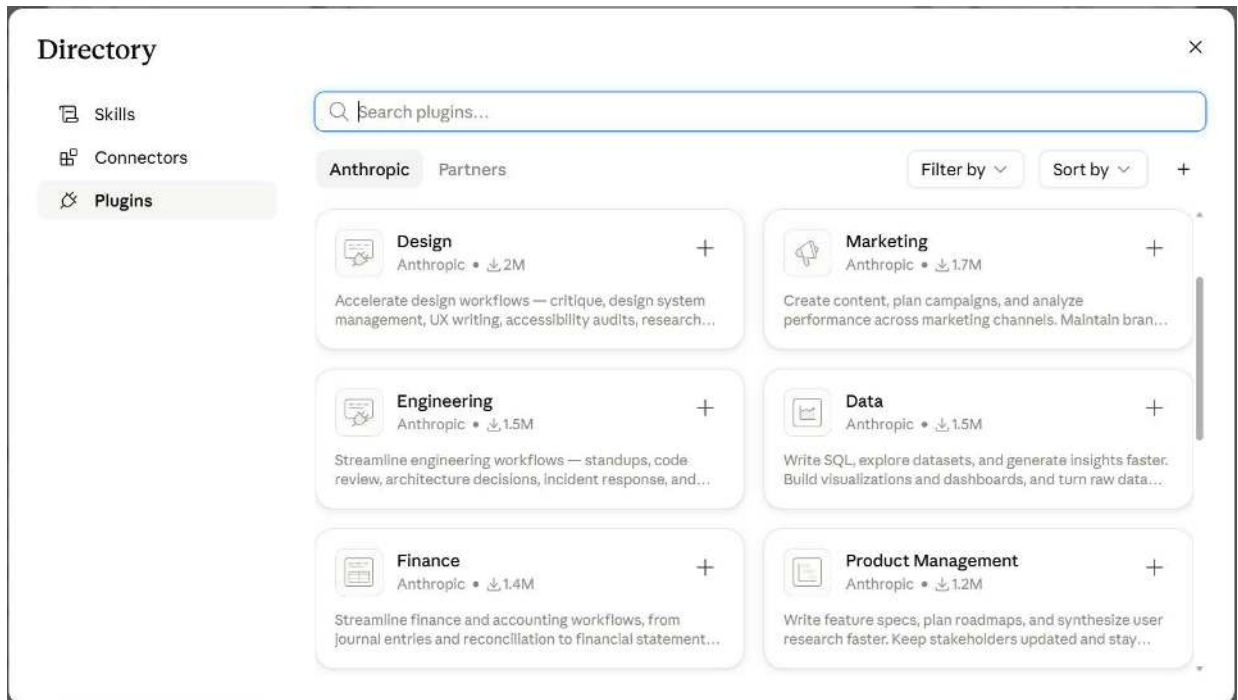


Figure 2.8E: The Plugins section of Directory shows plugins from Anthropic and Partners that you can install

8. Select a plugin that interests you, like **Marketing**, and note that a plugin combines skills and connectors:

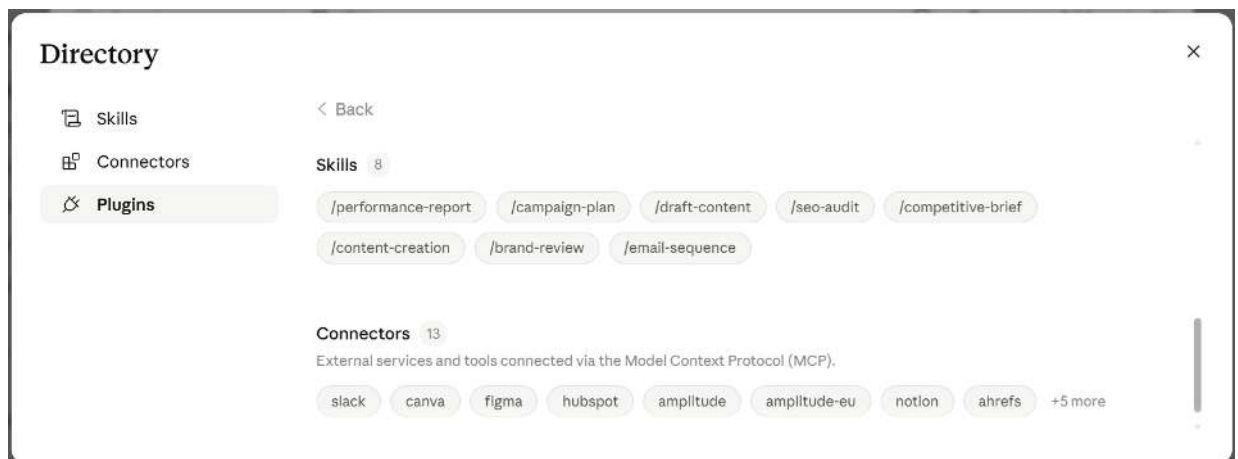


Figure 2.8F: Skills and connectors that will be installed with the Marketing plugin

9. Select **Connectors**, and note that they can be added (with the + button), configured (with the gear button), and reconnected (with the reconnect button):

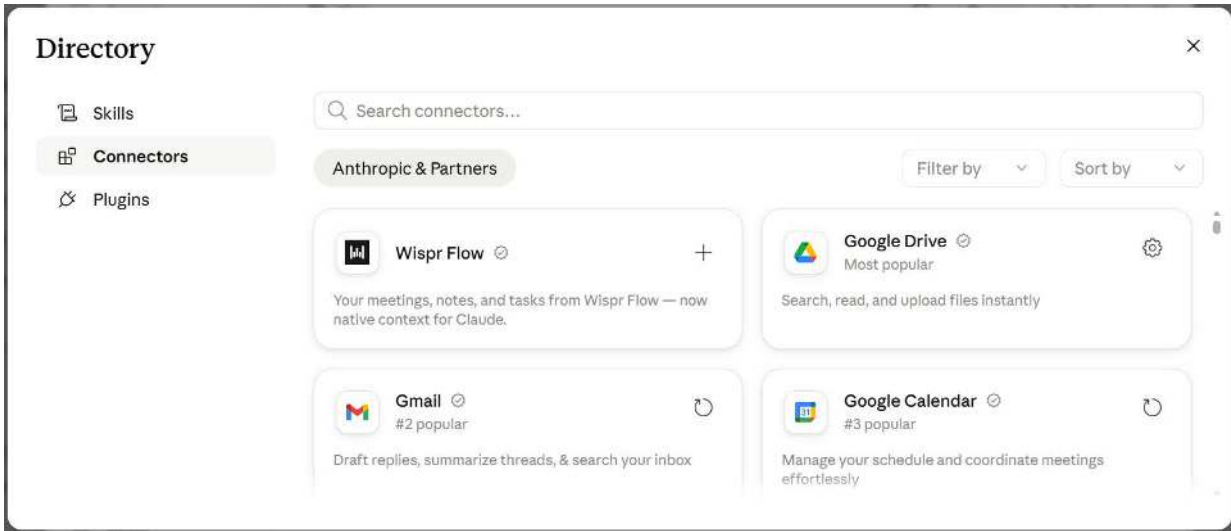


Figure 2.8G: Connectors can be added, configured, and reconnected

10. Select **Skills** and note preinstalled skills like /morning can be configured, and other skills can be added with the + button:

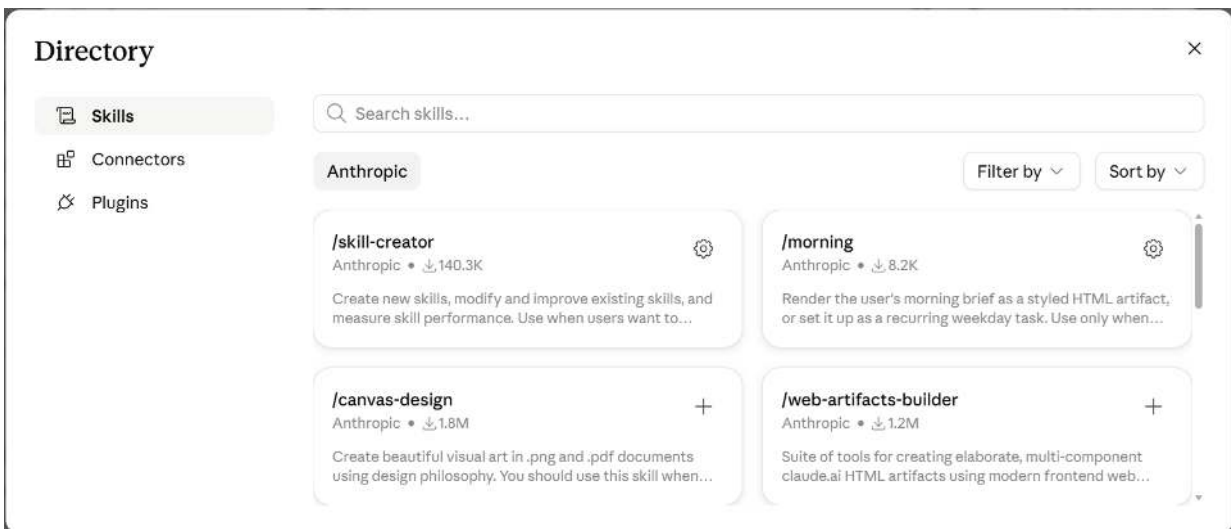


Figure 2.8H: Installed skills can be viewed and configured from Directory

11. Close **Directory**.
12. Close **Settings**.

You will learn about skills in detail in *Book 3*, including how to build your own custom skills.



Good practice: Install only what your task needs. You do not need to connect Google Drive, Microsoft 365, Slack, or another external service merely to work with files in Claude. You can upload a document, spreadsheet, presentation, or other supported file directly.

A plugin's listing should explain what the plugin helps you do, which skills it contains, which connectors it uses, whether they are required or optional, what setup or sign-in is needed, which Claude surfaces support it, and what permissions it may request.

Search for the result you want rather than installing everything that sounds interesting. For example, you might search for presentation creation, spreadsheet analysis, document editing, project reporting, or a particular service you already use.

A plugin containing only skills may not require an external account. A plugin containing a connector may ask you to sign in to another service and approve access.

Availability can depend on your Claude plan, device, region, workspace, and role. In a managed workplace or school account, an administrator may decide which plugins and connectors you can use. If an Install or Connect button is unavailable, the capability may be restricted by your organization.



Watch out: A listing in the Directory does not guarantee that every included capability is available to you. A required connector may still need to be enabled by an administrator or connected to your account.

Review connector access and permissions

Adding a plugin may allow Claude to read information or take actions in another service. The available controls depend on the plugin and your workspace.



Good practice: Read the authorization screen before approving it. Check which account you are connecting, what information the connector can access, and whether it can make changes.

When a plugin, skill, or connector can read files, reach another service, or take an action outside the chat, Claude will show what it is asking permission to do. Read this request before approving it, the same way you would read a permission request on your phone.



Watch out: Never approve a permission request you do not understand. Ask Claude to explain what the tool needs and why, in plain language, before you say yes.

When permission choices are available, prefer settings that ask before making changes. Reading a document is different from deleting it, sending it, sharing it, or replacing it.

Before connecting any account or service, check your employer's policy on which sources may be connected and who can approve the connection. Some organizations manage this centrally, while individual accounts may allow you to connect your own approved services directly.

Once a source is connected, describe what you're looking for in plain language, such as a project name, a date range, or the people involved. Claude searches the connected source and returns the files or messages that match.

Ask Claude to use only files from a specific folder, date range, or sender, rather than searching the whole connected source. A narrower request also produces a more focused, more useful result.



Watch out: Connecting a source gives Claude the ability to search everything within the scope you approve, not only the files related to your current task. Review that scope before connecting.



Learn more online: Use connectors to extend Claude's capabilities: <https://support.claude.com/en/articles/11176164-use-connectors-to-extend-claude-s-capabilities>.

Grant only the access a task genuinely needs. If a plugin asks for more than its stated purpose requires, do not connect it, or contact the service to ask why.



Watch out: A connected plugin can keep working even after you stop using it regularly. Review your connected plugins occasionally, not only when you first set them up.

Remove access that is no longer needed

When you finish a project or stop using a service, disconnect the plugin rather than leaving it connected. Removing access does not delete your work in the other service. It only stops Claude from reaching that service going forward.



Real-world example: An operations manager connects a project-management plugin for a three-month product launch. Once the launch ends, she removes the connection, since the plugin no longer has an active purpose and would otherwise keep standing access to the project data.

Select a plugin explicitly when necessary

Claude may recognize when an installed plugin is relevant. You can also select a plugin or one of its included capabilities before sending your request.

Depending on the Claude version you are using, you may be able to mention the plugin using @ followed by its name, select it from the attachment or + menu, select it as a source in Cowork, or open the plugin from the Directory.

An explicit selection is useful when more than one installed capability could handle the task, or you want Claude to use information from a particular service.

For example, you could prompt:

```
Using the Google Drive connector, find the latest approved project brief for the customer portal.
```

```
Summarize the agreed scope, deadlines, owners, and unresolved decisions. Include links to the source files. Do not use draft documents or files outside the approved project folder.
```

Naming the source and its boundaries reduces the chance that Claude will use an unrelated file.

You do not need to install every available document, spreadsheet, or presentation plugin. Claude may include preinstalled skills or plugins for common document, spreadsheet, and presentation tasks. Check Customize to see what is available in your account.

Cite and review source material

Ask Claude to name or link the specific file or message behind each claim it makes from a connected source. Open a sample of these citations yourself before relying on the result, especially for anything that will inform a decision.



Real-world example: An office manager connects a shared drive and asks Claude to find every vendor contract expiring in the next 60 days, then opens each cited file to confirm the expiration date before renewing anything.



Good practice: treat a connected search the same way you'd treat a colleague's summary of documents you haven't read yourself. Useful as a starting point, worth checking before you repeat it as fact.

Try it now

Open the plugin directory and find one plugin connected to a service you already use. Read its description and the permissions it would request, without connecting it yet. Decide whether you would approve that request and why.

Using an approved connected source, ask Claude to find and summarize the files or messages related to one specific project or topic. Limit the request to a folder or date range you choose.

Check the result

- ☐ Could you explain, in one sentence, what the plugin you found actually does?
- ☐ Did the requested permissions match the plugin's stated purpose?
- ☐ Do you know where to go later to disconnect a plugin you no longer need?
- ☐ Do you know what a permission request looks like before you meet one in a real task?
- ☐ Did Claude use only the sources you intended, based on the scope you set?
- ☐ Can you open the cited files and messages to confirm they say what the summary claims?
- ☐ Would you be comfortable sharing this result with someone outside your team, given what it draws on?



What to ask next: List every connected source you used to answer this, and tell me if any file you checked seemed outdated.

Connecting a source is only useful if Claude can act on what it finds. You've now learned every topic this chapter pulls together: drafting and revising documents, summarizing information, analyzing data, creating media, and adding plugins and connectors. *Chapter 2.9* asks you to combine them into one real, recurring briefing you'll actually use.

Chapter 2.9 Project – Produce a Useful Weekly Briefing

This project brings together everything from *Book 2*. You'll build one weekly briefing, gather its material from approved sources, and produce it with Cowork.

You need at least one approved source of information, such as connected files, a shared drive, or your own notes. By the end of this project, you'll have a working weekly briefing and a process for improving it over time.

Figure 2.9A traces the full path from your connected sources, through Cowork, to your approval, and finally to the finished briefing:

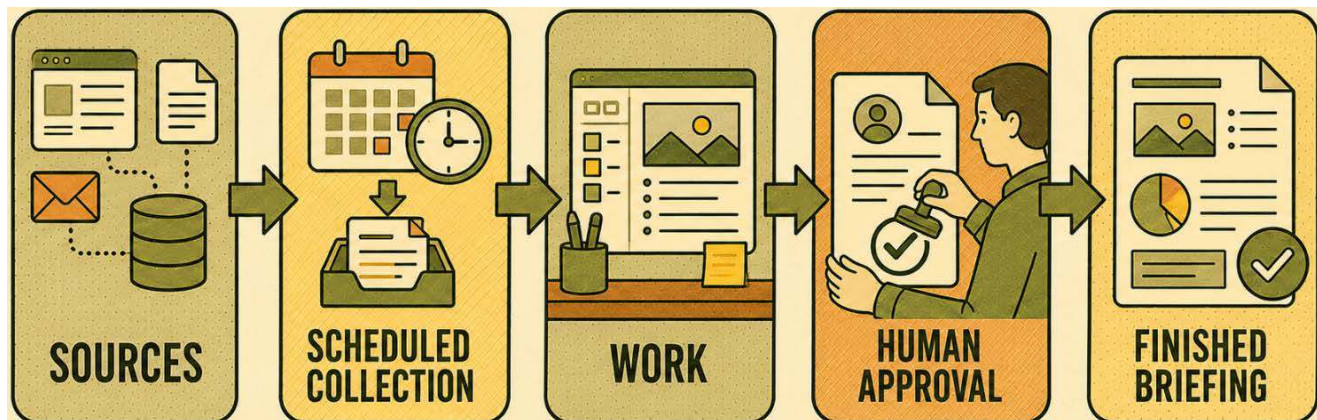


Figure 2.9A: Your approval sits between Cowork's draft and the finished briefing every week, not just on the first run

Fictional worked example

Brightside Learning Weekly Operations Briefing continues the fictional organization used in the spreadsheet and presentation chapters, while bringing together files, evidence, decisions, and human review.



Sample files: You can download files that all start with 2-9 from the following folder:
<https://github.com/markjprice/claude-vb/tree/main/Files/Book2/>.

Use these files in sequence:

- 2-9-Approved-Source-Register.docx: Defines which sources may be used, what each source can support, and what information is excluded.
- 2-9-Weekly-Workshop-Tracker.xlsx: Contains an editable dashboard, current-week workshop results, calculated attendance rates, and the next-week schedule.
- 2-9-Team-Update-Notes.docx: Supplies confirmed risks, recommendations, client follow-ups, owners, and deadlines.
- 2-9-Previous-Weekly-Briefing.docx: Provides a model for structure and tone. Its old figures are deliberately not approved as current evidence.
- 2-9-Example-Weekly-Briefing.docx: Shows the one-page result Claude should produce after gathering and checking the evidence.

The files include several useful details:

- Attendance must be calculated from total attendees and total capacity
- One satisfaction score is missing because its survey remains open
- A lower result has no confirmed explanation
- One session is full and has a waitlist
- One instructor arrangement is still at risk
- One decision needs management approval
- The previous briefing may guide presentation but not supply current facts

In this project, you will produce a weekly operations briefing for Brightside Learning, a fictional training business. The managing director and program manager use the briefing to understand the past week, prepare for the next one, and make decisions.

Your approved sources include a workshop tracker, team update notes, and the previous approved briefing. You will define what the audience needs, gather evidence from those sources, create a one-page briefing, check every important claim, and prepare the next edition.

You can adapt the same process for a family planning update, project report, client briefing, community newsletter, learning review, or small-business summary.

Define the audience and reporting questions

A useful briefing is written for a particular group of people. It should help them understand, decide, or do something. “Summarize this week” gives Claude too little direction.

Begin by naming the audience and the questions the briefing must answer. Also explain what the audience already knows and how much time they have.

Upload 2-9-Approved-Source-Register.docx, and use this prompt:

```
I am creating a weekly operations briefing for Brightside Learning.
```

```
Audience:
```

```
The managing director and program manager, who know the workshop business but haven't seen this week's results or team notes.
```

```
Purpose:
```

```
Help them understand this week, prepare for next week, and make any decisions that can't wait.
```

```
Keep it to a three-minute read, then create a briefing plan (not the briefing itself) that states:
```

- What this audience needs to understand
- Decisions or actions to support
- Five reporting questions to answer
- Evidence needed for each question
- Information to leave out
- Length, tone, and structure
- Any rule that could prevent a misleading result

A suitable response should identify questions such as:

- How many sessions were delivered?
- What were overall attendance, known net revenue, and satisfaction?
- Which results were unusually strong or weak?
- What risks or decisions affect next week?
- What information remains missing?

Review the questions before gathering information. Remove questions that are interesting but do not help this audience act. Each reporting question should have a reason for being included and at least one approved source that can answer it.

Gather information from approved sources

The source register gives Claude boundaries. It identifies what may be used, what must be excluded, and how evidence should be labeled.

Upload the tracker, team notes, and previous briefing. Ask Claude to gather evidence before drafting prose:

Use the approved briefing plan and these attached files:

- 2-9-Approved-Source-Register.docx
- 2-9-Weekly-Workshop-Tracker.xlsx
- 2-9-Team-Update-Notes.docx
- 2-9-Previous-Weekly-Briefing.docx

Read the Source Register first and follow its rules. Do not write the finished briefing yet.

For each reporting question, build an evidence table with these columns:

- Reporting question
- Finding
- Exact figure or confirmed fact
- Source filename
- Sheet, row, cell, or heading
- Limitation or missing information
- Suitable for the briefing: Yes, No, or Needs review

Calculation rules:

- Calculate overall attendance as total attendees ÷ total capacity, not an average of session percentages.
- Include known net revenue from delivered sessions only.
- Exclude missing satisfaction scores from the average, and state how many sessions have a score.

Source rules:

- Use the previous briefing for structure and tone only, never as current figures.
- Don't guess why a result changed, or invent forecasts, revenue, client details, owners, or deadlines.

- Mark conflicting or unsupported information with [CHECK].

Finish with:

1. Three most important findings
2. Decisions or actions needing attention
3. Risks or missing information to disclose
4. Any question the approved sources can't answer

This intermediate table is important. It separates gathering evidence from writing persuasive prose. It also gives the reader a chance to catch a calculation error or unsupported conclusion before it reaches the briefing.

In the supplied example, Claude should find:

- Four delivered sessions
- Attendance of 58 out of 72 places, or 80.6%
- Known net revenue of £11,340
- Average satisfaction of about 4.5 from three of four sessions
- A full AI for Office Tasks session with four people waiting
- A pending backup instructor confirmation
- One management decision and two follow-up actions
- No approved explanation for Project Planning's lower attendance



Watch out: Missing information is not the same as zero. An open satisfaction survey should be reported as missing or pending.

Produce the briefing and supporting evidence

Once the evidence table is accurate, ask Claude to turn it into a short briefing. The main briefing should remain readable, while its source labels make important claims easy to check:

Using the approved briefing plan and the checked evidence table, create the Brightside Learning Weekly Operations Briefing for the week ending July 31, 2026. Return it as an editable Word document.

Audience: The managing director and program manager. Keep the complete briefing to one page, about a three-minute read.

Use these sections:

- A message-based title
- In brief
- What changed this week
- Decision and actions
- Risks and missing information
- Supporting evidence

Content requirements:

- State delivered sessions, total attendance, and overall attendance rate.
- Report known net revenue from delivered sessions only, and average satisfaction with its coverage.
- Identify the strongest and weakest attendance results, without inventing an explanation.
- State the decision needed on the proposed overflow session.
- Give each confirmed action an owner and due date.
- Disclose the missing satisfaction result and pending instructor confirmation.

Evidence requirements:

- Support every important figure or claim with a cited, approved source (filename plus sheet, cell, row, or heading).
- Use the previous briefing for structure and tone only, and preserve uncertainty from the sources.
- Mark anything needing confirmation with [CHECK].

Design requirements:

- Use a clean one-page layout with the summary and required decision easy to find.
- Use a small table for decisions and actions, with text large enough for comfortable reading.
- No decorative images or unsupported charts.

Do not send, publish, or distribute the briefing. Return it for human review.

The supporting evidence does not need to interrupt every sentence. A short evidence section can identify the workbook sheets, cell ranges, and note headings used. For a higher-risk or client-facing report, use numbered source markers beside important claims and provide a fuller evidence appendix. A reader should be able to distinguish results, decisions, recommendations, risks, and missing information without guessing.

Review the briefing

A fluent briefing can still contain a wrong number or unsupported explanation. Review it against the original sources rather than checking only whether it sounds convincing.

Upload the generated briefing alongside the source files:

Review the attached weekly briefing against:

- 2-9-Approved-Source-Register.docx
- 2-9-Weekly-Workshop-Tracker.xlsx
- 2-9-Team-Update-Notes.docx
- 2-9-Previous-Weekly-Briefing.docx

Do not edit the briefing yet.

Check every important claim for:

- A matching approved source, with correct arithmetic and reporting period
- Correct owner and deadline, and any missing qualification or uncertainty

- Unsupported causes, forecasts, or recommendations
- Accidental reuse of figures from the previous briefing
- Confidential, unnecessary, or misleadingly presented missing values
- A source citation that doesn't lead to the stated evidence

Create a review table with these columns:

- Briefing claim
- Source checked
- Result: Supported, Incorrect, Incomplete, or Unsupported
- Problem found
- Recommended change
- Priority: High, Medium, or Low

Also confirm the briefing stays on one page, leads with the most important information and a clear decision, gives every action an owner and date, and is ready for human approval.

Do not change the document until I approve the review.

After checking the review table, the reader can approve specific revisions:

Apply only the High-priority changes that I approved.

Do not change any other figure, conclusion, wording, source citation, owner, deadline, formatting, or layout.

Return the revised briefing as a new Word document so the previous version remains available.

Try it now

Build a first draft of your briefing by hand, using the sources and questions you defined. You can choose one of three versions of this project:

1. A personal or family planning briefing
2. A team, manager, or executive briefing
3. A client, market, or small-business briefing

Pick the version that matches a briefing you'd actually use, since a real audience gives you a genuine reason to keep improving it.

Check the result

- ☐ Does the briefing answer the specific questions you defined, without drifting into unrelated material?
- ☐ Can you open the linked sources and confirm they support the summary?
- ☐ Would your intended audience find this briefing worth reading every week?

The next step

Your weekly briefing is a real result, but right now it still depends on you: checking in, adjusting its questions, and deciding when it's ready to run on its own. *Book 3, Automate Recurring Tasks*, picks up exactly there. You'll learn to turn a working routine like this one into a skill or a scheduled task that Claude can run without you standing over it, and how to keep control over what it's allowed to do along the way.

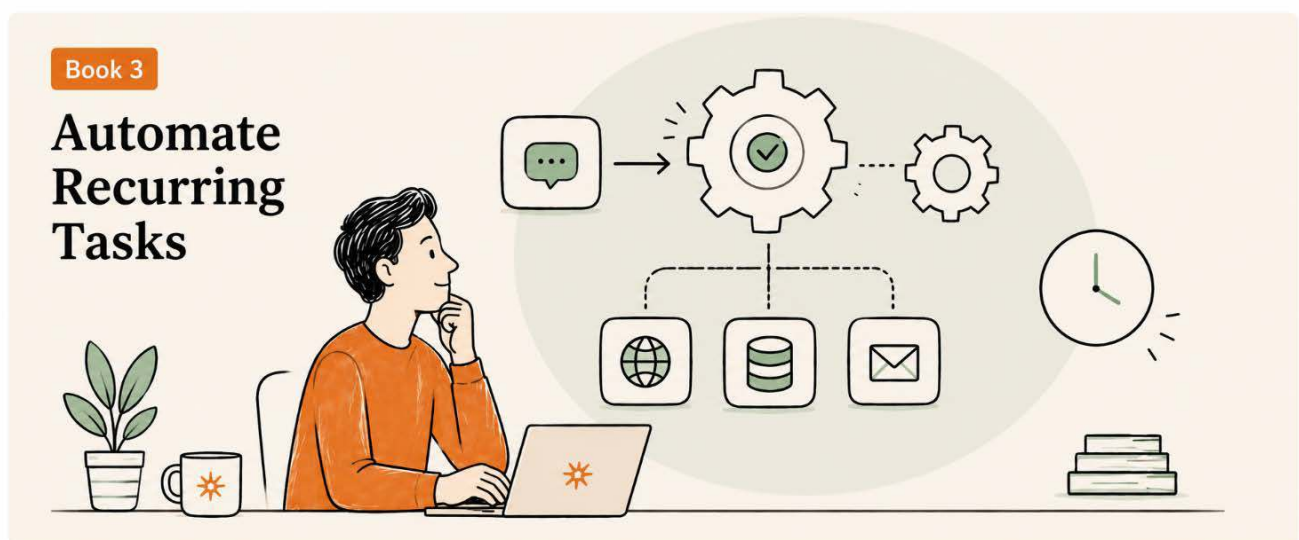
Book 3

Automate Recurring Tasks

Book 3 teaches you how to turn work you repeat into a reusable skill or scheduled Cowork task, connect Claude to trusted information and tools, and, when no better integration exists, let Claude operate software through computer use.

It has eight chapters, including an end-of-book project:

- 3.1 Choose the Right Automation Building Block 126
- 3.2 Schedule Recurring Work 128
- 3.3 Turn a Procedure into a Reusable Skill 135
- 3.4 Build a Library of Skills for Real Work 149
- 3.5 Connect Trusted Information and Tools with MCP 155
- 3.6 Control Software with Computer Use 162
- 3.7 Automate Personal Organization and Build Workflows 172
- 3.8 Project – Build and Measure an End-to-End Workflow 177



Chapter 3.1 Choose the Right Automation Building Block

By now you have used Claude for single conversations, research, and finished documents. This chapter is about the next step: work you do again and again. You will learn to spot that kind of work, and to pick the simplest tool that handles it well. You do not need to memorize every option today. You need a way to decide, each time a task repeats, what to build.



Watch out: Availability varies by feature. Cowork and scheduled tasks require a paid Claude plan. Skills are available on all plans when code execution is enabled. Computer use currently has narrower plan support.

Recognize repeated and multi-step work

Some tasks happen once. Others come back every week, every project, or every new client. A task is worth automating when it repeats, follows a pattern, and takes time to redo by hand.

Watch for phrases you say to yourself, such as “I did this last month too” or “I always have to remind Claude about our format.” Those are signs that a one-off conversation should become something more permanent.

A multi-step task has several stages that must happen in order, such as gathering information, drafting a document, and checking it against a rule. These tasks benefit the most from structure, because a plain chat can lose track of the stages.



Watch out: Do not automate a task you have only done once. Without a few repetitions, you cannot yet tell which parts are stable and which parts change each time.

Compare projects, scheduled tasks, skills, connectors, and plugins

Claude offers several ways to make repeated work easier. Each one solves a different problem:

- A **project** keeps related chats, files, and instructions together. Use it when work is repeated but still needs your judgment at each step. You were introduced to projects in *Chapter 1.11*.
- A **scheduled task** stores a Cowork task so Claude can run it automatically on a recurring schedule or whenever you start it on demand. Use it when the same well-defined work should happen repeatedly without rebuilding the instructions each time. You will learn about scheduled tasks in *Chapter 3.2*.
- A **skill** is a folder of reusable instructions, and optionally examples, resources, and scripts, that Claude loads when a task needs that specialized procedure. Use it when the steps are stable and well understood. You will learn about planning, building, and testing skills in *Chapter 3.3*. You will then build a library of skills and learn how to share them in *Chapter 3.4*. Skills are not limited to paid plans. Anthropic currently makes them available on free as well as paid plans when code execution is enabled.
- Some tasks need more than instructions. They need current information from another system, such as a calendar, a shared drive, or a company database. When that happens, a connector or a **Model Context Protocol (MCP)** connection brings that outside system into the conversation safely, with permissions you control. You will learn about custom connectors and MCP in *Chapter 3.5*.

- A **plugin** packages several related capabilities, such as skills and connectors, so Claude can be customized for a broader role or workflow. Use a focused skill for one procedure; use a plugin when several reusable capabilities belong together.

To guide you in choosing the right automation building block for your work, use the matrix shown in Figure 3.1A:

	How Often It Repeats			How Specialized			Outside Information			Complexity			Minimum Plan
	Low +---+	Med +---+	High +---+	Low +---+	Med +---+	High +---+	Low +---+	Med +---+	High +---+	Low +---+	Med +---+	High +---+	
 Chat	●			●			●			●			Free
 Project		●			●			●			●		Free
 Scheduled Task			●	●				●			●		Pro
 Skill			●			●		●			●		Free
 MCP			●			●			●			●	Free

Figure 3.1A: Match your task to a building block by asking how often it repeats, how specialized it is, how much outside access it needs, and its complexity



Good practice: Add an MCP connection only when the task genuinely depends on outside data or action. A connection you do not need adds risk without adding value.

A durable progression for building automation

Use this order whenever you consider automating a task. It keeps you from building something complicated before you know the task well:

1. Have Claude help with one task.
2. Repeat and refine the task or procedure manually.
3. Store its context in a project.
4. Schedule or automate only after the procedure is reliable.
5. Turn stable instructions into a reusable skill.
6. Connect approved information or tools.
7. Measure time, quality, error rates, and exceptions.
8. Package related skills and connectors into a plugin if the workflow grows beyond one procedure.



Good practice: The core question at every stage is the same: what is the least complicated mechanism that reliably handles this work? Start with the plainest option that could work. You can always add a skill, a connection, or a subagent later, once you know exactly where the plain version falls short.

Try it now

Pick one task you have asked Claude to help with more than twice in the last month. Write down each stage of that task, in order. Then use the questions in this chapter to guess which building block fits best. You will build it in the chapters ahead.

Check the result

- ☐ Did you name a task that truly repeats, rather than a one-off request?
- ☐ Can you list its stages in order?
- ☐ Does your guess about the right building block match the reasons given in this chapter, not just the newest or most advanced option?

Choosing the right building block is only the first decision. Before you build anything, *Chapter 3.2* covers turning a manual routine into a task that runs on a schedule or a trigger.

Chapter 3.2 Schedule Recurring Work

Scheduling is useful when the problem is not how to do a task, but remembering to do it at the right time. Claude does not need to wait for you to open a conversation and repeat the request. You describe the work once, choose when it should run, and Claude creates a fresh Cowork run at that time. This works especially well for briefings, reports, research checks, and other work where the inputs change but the procedure stays mostly the same.

Cowork can save a task and run it automatically on a recurring schedule or keep it ready for you to run on demand. This chapter covers creating a scheduled task, choosing a cadence, reviewing past runs, and pausing, editing, or deleting tasks.

Figure 3.2A shows how a time-based trigger leads to a task running, checking a source, producing a result, and sending you a notification:

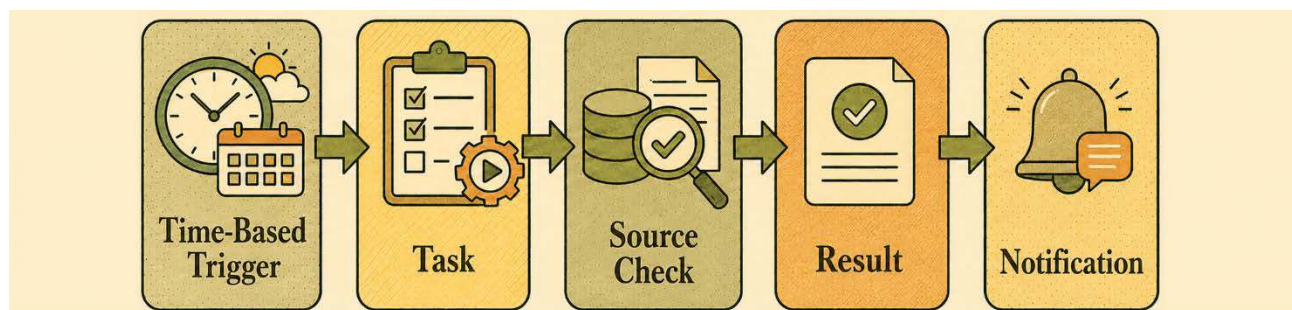


Figure 3.2A: Every scheduled task follows this same chain. If the notification stops arriving, check each stage to find where it broke

A scheduled task has two parts:

1. **The work:** What Claude should do
2. **The trigger:** When Claude should do it

The work itself can also contain a condition. For example, Claude might check once each day but notify you only when tickets become available for a music concert that you want to attend.

To complete this chapter, you need a task you already do by hand on some kind of repeating basis, such as a weekly check or a monthly report. Think about any of your work that follows one of these patterns: “I do this every Monday,” “I check this website every few days,” “I prepare the same report each month,” “I need to remember this on a particular date,” or “I’m waiting for something to change.”

The best first task is small, useful, and easy to check. I discourage you from beginning with anything that automatically spends money, sends messages, submits forms, deletes information, or makes an important decision!

If you can’t think of anything, you can use mine, which I like to run every Thursday evening (and just adapt it to your own location):

```
Find five interesting things happening in or near Brighton this weekend.
```

By the end of this chapter, you’ll be able to turn that manual routine into a scheduled task, including one that watches for a change.



Good practice: Before scheduling recurring work, try the prompt without a trigger once in a normal chat. Check whether the result contains the right information and level of detail. Scheduling an unclear prompt only repeats the unclear result.

Approving a scheduled task

Scheduled tasks may need permission to use tools or take actions. The approval mode controls how Claude handles those permissions when you are not there watching the task.

For the Brighton events example, **Auto** is appropriate because the task only researches public information and prepares a private report. It does not send messages, spend money, modify records, or publish anything.

Other approval choices include **Manually approve** and **Skip all approvals**.



Watch out: Do not choose the most permissive approval mode simply to avoid interruptions. Match the approval setting to what the task is allowed to do.

Create your first scheduled task

The easiest, most controlled way to schedule a task is to use a simple dialog box on the **Scheduled** page:

1. In the Claude sidebar, select **Scheduled**.

2. Select **New task** and then select **Set up manually**. This provides a simple interface and direct control of the task:

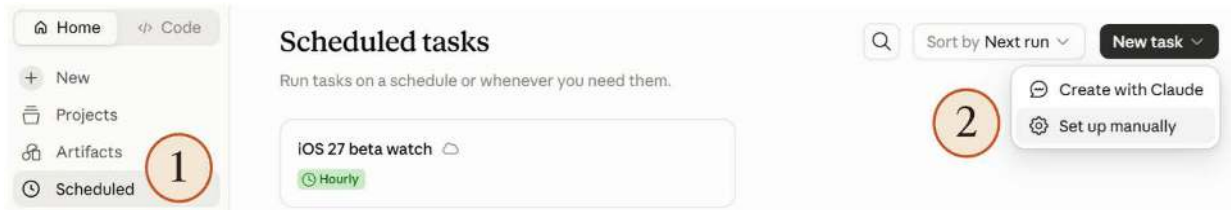


Figure 3.2B: Creating a scheduled task with a simple user interface

3. Define the task name.
4. Define the prompt describing the task:

Find five interesting things happening in or near Brighton this weekend.

For each event, include:

- Event name
- Date and time
- Location
- Admission price, or "Free"
- A link to the official event page
- One sentence explaining who might enjoy it

Use current information from official event or venue websites. Do not include an event if its date or availability cannot be confirmed.

5. Set the approval mode to **Auto**.
6. Set the schedule to run every Thursday at 6:00 p.m.:

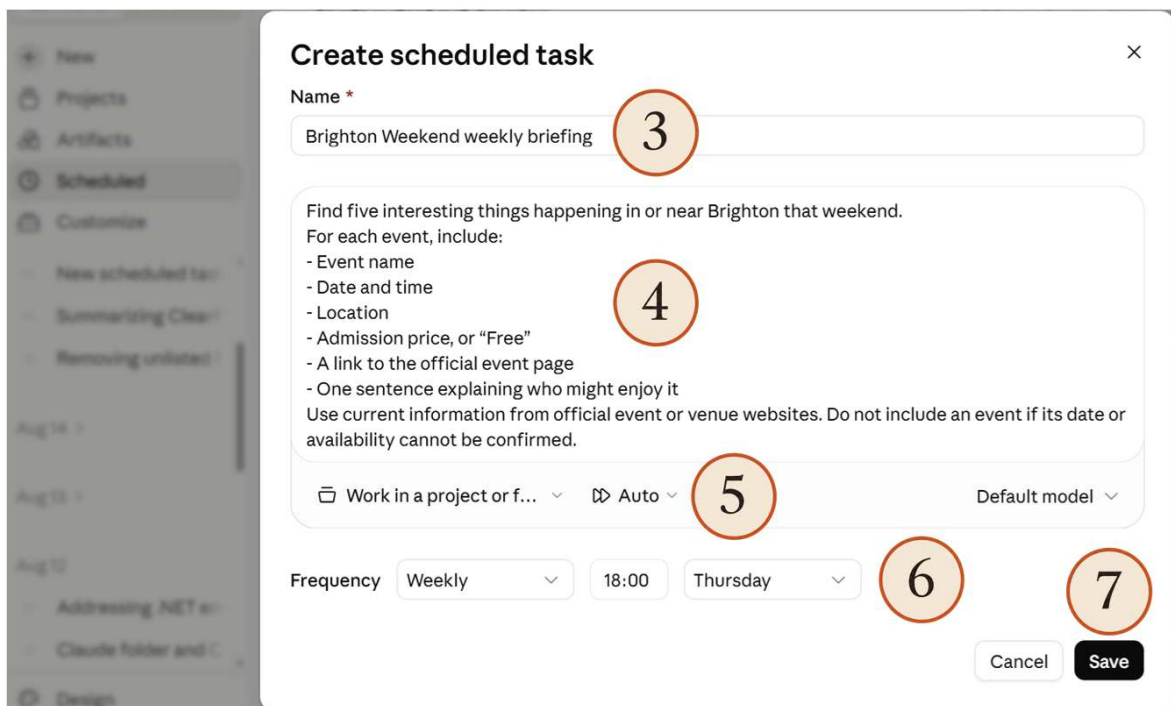


Figure 3.2C: Creating a scheduled task for a Brighton Weekend weekly briefing

7. Select the **Save** button.
8. Confirm it appears on the **Scheduled** page.
9. Note that you can run the task now, and pause, edit, or delete the task:

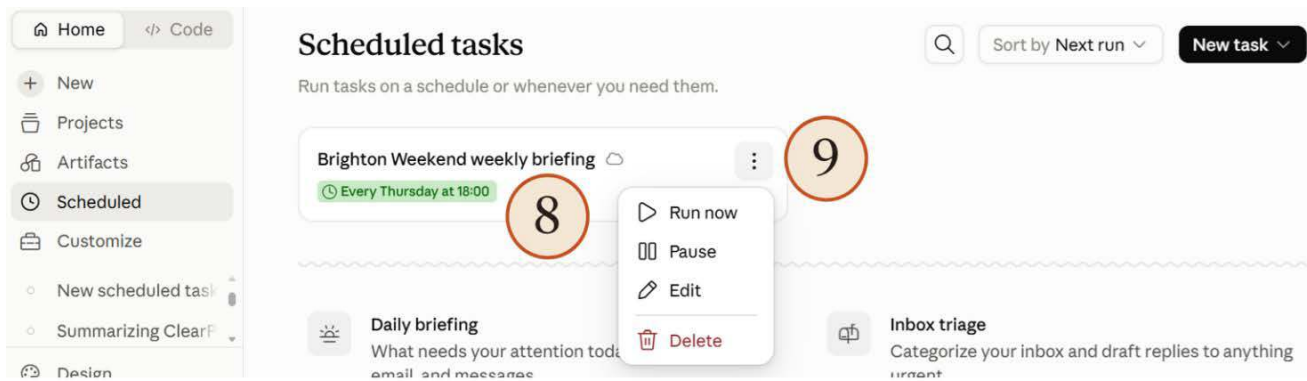


Figure 3.2D: You can run a task now, and pause, edit, or delete a task

Understand cloud and local dependencies

The cloud icon next to the scheduled task in Figure 3.2D shows that Claude can schedule the task remotely. Whether it can complete every step without your computer depends on the information and tools the task needs. The **Brighton Weekend weekly briefing** example scheduled task does not need any local computer resources so it can run independently in the cloud.

Most scheduled tasks normally run in Claude’s cloud. This means a task that uses the web, files saved with your Claude account, or supported cloud connectors can run even when your computer is asleep and your Claude desktop app is closed.

Some tasks, however, need something that exists only on your computer. Examples include a local file or folder, a local connector, your browser, or a desktop application. Claude can reach some of these resources through Claude desktop app, but your computer must be available and the desktop app must be running when Claude needs them.

Think about the difference this way:

Task needs	Computer required?	Example
Web information	No	Find events happening this weekend
Files stored with Claude	No	Summarize an uploaded weekly report
Cloud connector	Usually no	Summarize new messages from a connected service
File or folder on your computer	Yes, when Claude needs it	Process files in a local Reports folder
Local connector	Yes	Read information from software connected through your desktop
Browser or desktop app	Yes	Update information through an application on your computer

Table 3.2A: Cloud versus local scheduled tasks

If a scheduled task depends on something local, closing your laptop can prevent that part of the task from working even though the scheduled task itself still exists. The Claude desktop app has a setting to keep your computer awake but closing a laptop lid will still put it to sleep:

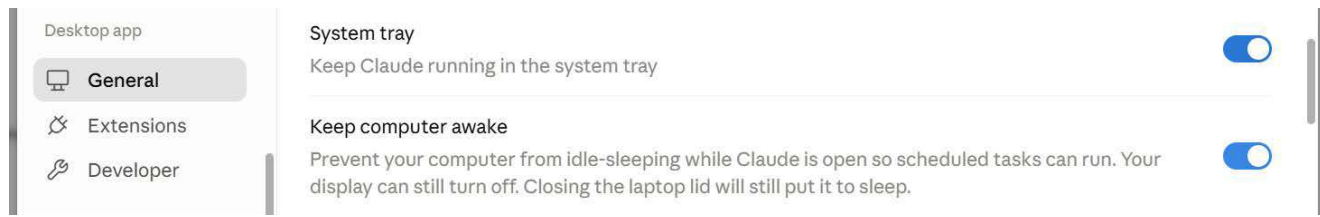


Figure 3.2E: Keep your computer awake so that local scheduled tasks can execute

Do not assume that because a task appears under **Scheduled**, everything it needs will still be available when it runs. Ask yourself, “Where does this task get its information, and will that source still be reachable at the scheduled time?”



Good practice: Prefer cloud-accessible sources for work that must run reliably while you are away from your computer. Use local resources when the task genuinely needs information or software that exists only on your machine.

Alternative ways to create a scheduled task

The **New task** button has two ways to create a scheduled task: **Set up manually** and **Create with Claude**. **Create with Claude** is useful when you know the outcome you want but are unsure how to describe the schedule. Claude asks questions until it has enough information to create the task. **Set up manually** is quicker when you already know the prompt and schedule you want.

To trigger the interactive experience from the chat box, type the slash command `/schedule` and then answer any questions from Claude.

Include the following details in the prompt when they matter:

- What should Claude do?
- When should it run?
- Where should it get its information?
- What should the result contain?
- When should it produce or withhold a result?
- What must it never do without you?

At the scheduled time, Claude searches for information and prepares the requested list.

Choosing a cadence and providing resource context

Choose a time that gives the task access to fresh information, such as after a weekly report normally arrives. Each scheduled run should be able to find the information it needs. Do not assume that Claude will remember a value merely because the previous run mentioned it.

For example, avoid:

```
Compare this week's figures with the figures you found last week.
```

Instead, prefer:

Find this week's and last week's figures from the approved source, compare them, and report the change.

If a team uploads its weekly figures by 4:00 p.m. Friday and you need a summary for a 9:00 a.m. Monday meeting, schedule the task for Friday evening or early Monday morning, not Friday at noon.



Watch out: Check which time zone Claude is using before scheduling something tied to a deadline or business day, especially when you travel or work with people in another country. “Friday at 6:00 p.m.” is only unambiguous if you know whose 6:00 p.m. you mean.

If the task needs lasting context, store that information somewhere Claude can reach reliably, such as the relevant project, files saved with Claude, or a connected service.

Recurring work often needs updated information. Make sure the task can still reach its sources when it runs. A file stored only on your computer may not be available to a web task. A connector may also stop working if its permission expires or your organization disables it.

Review the first few results. Adjust the prompt if the task returns too much information, repeats unimportant details, or misses the reason you scheduled it.



Good practice: Begin with a weekly schedule unless the information changes quickly. A daily task that rarely finds anything useful soon becomes noise. Name each scheduled task clearly enough that you recognize its purpose months later, without needing to open it to remember what it does.

Schedule inside a project when the task needs context

A standalone scheduled task works well when its prompt and sources contain everything it needs. If several recurring tasks belong to the same piece of work, consider putting them in a Cowork project instead.

A project can provide instructions, context, memory, and scheduled tasks that belong together. For example, a project called Weekly Marketing Review could contain your reporting rules and source material, with a scheduled task that prepares the review every Friday.



Good practice: Use a project for the information and continuing context of the work. Use the scheduled task for deciding what should happen repeatedly and when.



Learn more online: *Schedule recurring tasks in Claude Cowork:* <https://support.claude.com/en/articles/13854387-schedule-recurring-tasks-in-claude-cowork>.

Design silence carefully

Some monitoring tasks are supposed to say nothing when there is no change. That creates a problem: silence can also mean that the task failed.

For anything important, decide how you will distinguish those cases. Review the run history periodically, or ask the task to produce a short status result such as “Checked successfully; no change found” if receiving regular confirmation is more useful than silence.

Review, pause, edit, and delete tasks

Once a month, check four things: Useful? Accurate? Working? Still safe? Then pause or delete tasks you no longer need, repair failed tasks, revise noisy prompts, and reconsider any task whose permissions or data sources have changed.

Pause a task when you may need it again but do not want it to run now. For example, pause a weekly job search while you are not looking for work. A paused task keeps its instructions but does not run. Resume it when you are ready. Check the next run after resuming because the original time may already have passed.

Edit a task when its purpose remains useful, but its instructions or timing need to change.

Delete a task when you no longer need its instructions or run history. Deleting a task stops future runs. Pause before deleting if you are unsure. Pausing is easier to reverse.



Real-world example: A job seeker sets up a weekly task that checks three job boards for new postings matching a specific title and location, and gets a summary only when a new match appears.



Good practice: A scheduled task can drift away from its original purpose without anyone noticing and keep running while producing results nobody reviews. Review your scheduled tasks once a month. You could even create a scheduled task to remind you to do this! Pause or delete tasks that no longer help, repair tasks with failed runs, and revise tasks that create repeated noise.

Do not confuse a schedule with a skill:

- A scheduled task answers: When should Claude do this?
- A skill answers: How should Claude do this consistently?

Later, you can combine them: a scheduled task can run regularly while using a skill to perform the procedure the same way each time.



Good practice: Do not rely on a scheduled task for an emergency, medical warning, financial trade, legal deadline, or safety alarm. Start with low-risk summary and research tasks. Avoid scheduled tasks that send messages, make purchases, or take hard-to-reverse actions.

Try it now

Pick one routine check you currently do by hand, such as reviewing a report every Monday or watching for a change on a webpage.

Check the result

- ☐ Did the scheduled task run at the time and frequency you intended?
- ☐ Does the result match what you'd have found by checking manually?
- ☐ Would you notice quickly if this task stopped running or started producing an unhelpful result?

With scheduled tasks under your control, you're ready to build your first automation. *Chapter 3.3* shows you how to turn a procedure you already repeat by hand into a reusable skill.

Chapter 3.3 Turn a Procedure Into a Reusable Skill

In *Chapter 3.1*, you learned that a skill stores a tested procedure so Claude can repeat it consistently. This chapter shows you how to prepare a procedure before you build it. I will then show you how to create and test the finished skill.

Good preparation matters more than the building step itself. A skill built from a vague procedure will behave inconsistently, no matter how carefully you set it up.



Watch out: Skills are available on all plans, including Free, but need code execution to be enabled.

Enabling code execution

Skills require Claude's code execution capability. If the **Skills** section is missing, check **Settings | Capabilities** and make sure code execution and file creation are enabled:

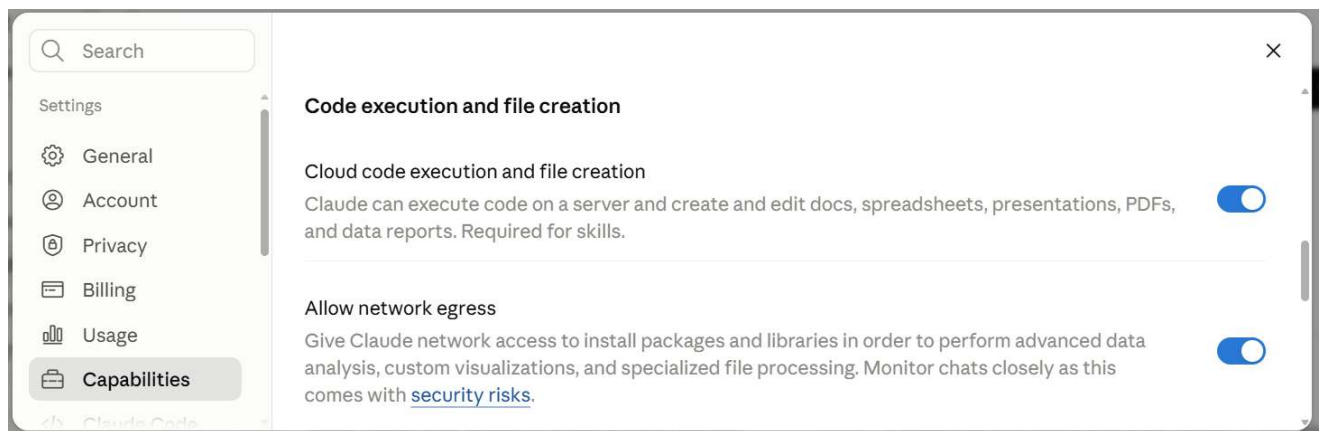


Figure 3.3A: Make sure Cloud code execution and file creation is enabled

In a Team or Enterprise organization, an administrator can control whether these capabilities are available so they might have disabled skills for you.

Understand what a skill really is

A skill is a small package that teaches Claude how to do a particular kind of work. Anthropic describes a skill as a folder containing instructions, and optionally scripts and other resources, that Claude can load when a task needs them.

The simplest skill might contain only a few paragraphs of instructions. A more advanced skill might also include a template, a worked example, a reference file, or code that Claude can run.

Imagine that you regularly ask Claude to review a client proposal. Without a skill, you might have to repeat the same instructions to Claude each time:

1. Check that every claim is supported.
2. Flag missing prices or dates.
3. Use our company terminology.
4. End with a short list of questions for the author.

A skill stores those instructions once. The next time you give Claude a proposal, Claude can load the skill and apply the same procedure.

This makes skills different from ordinary prompts. A prompt tells Claude what to do in this conversation. A skill teaches Claude a reusable capability that can be available in many conversations.

Skills are especially useful when:

- You repeat the same procedure with different inputs
- The procedure contains rules you do not want to explain every time
- Consistency matters across several people or projects
- You have examples or templates that improve the result
- The task is narrow enough to describe clearly

A skill is less useful when the work changes completely every time, when you are still discovering the procedure, or when the task depends mainly on judgment that cannot be turned into stable rules.

Figure 3.3B shows the basic idea:

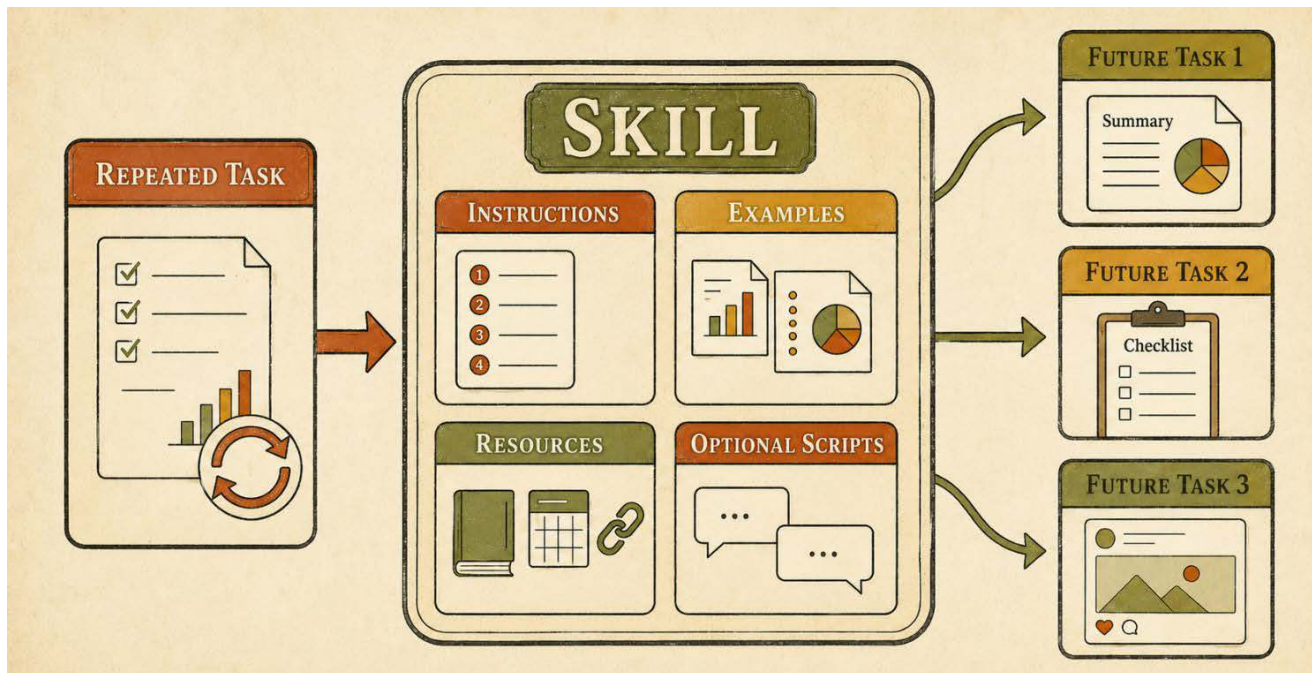


Figure 3.3B: A skill is a reusable capability that can be available in many conversations



Good practice: Make a skill small enough that you can explain its purpose in one sentence. Several focused skills are easier to test and combine than one enormous skill that tries to do everything.

How Claude decides when to use a skill

You do not normally need to type a special command every time you want a skill. Claude looks at the descriptions of the skills that are enabled for you, decides which ones are relevant, and loads the instructions it needs. That is why the skill description is more important than it first appears.

Let's see an example by comparing two skill descriptions.

First, a poor description that does not help Claude decide when to use the skill:

```
Helps with documents.
```

Second, a better description that gives Claude more details it can use to recognize situations where it could use the skill:

```
Reviews client proposals for missing facts, unsupported claims, pricing gaps, and required approval language.
```

The second description gives Claude a much clearer signal about when the skill should be used.

You can still be explicit when you want to make sure a particular skill is applied in a prompt:

```
Use my Client Proposal Review skill to review the attached proposal.
```

Skills installed through plugins can also appear in the / menu or the + menu, depending on where you are working in Claude.

Skills use what Anthropic calls **progressive disclosure**. Claude first sees enough information to decide whether a skill applies. It loads the fuller instructions and resources only when it needs them. This helps keep unrelated instructions out of the way.



Watch out: A skill does not remember the history of a project by itself. Put long-running context, files, and project-specific instructions in a project. Use a skill for the reusable procedure that can be applied to many projects.

Start with Anthropic's built-in skills

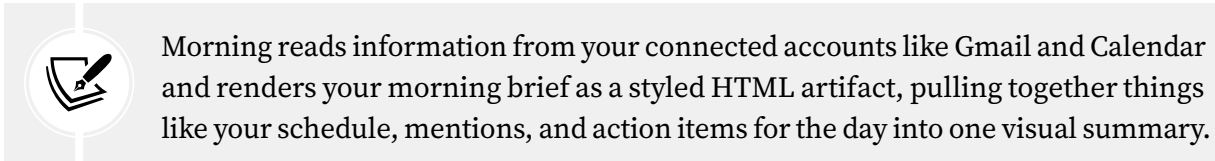
Before you build your own skill, try one that Anthropic already provides. Claude currently includes built-in skills for common document work, like Excel spreadsheet, Word document, PowerPoint presentation, and PDF document creation and manipulation.

Claude uses these automatically when they are enabled and your request needs them. For example, if you ask Claude to create a PowerPoint presentation, it can load the PowerPoint skill without you having to name it. You saw examples of this throughout *Book 2*.

These built-in skills are useful for two reasons. First, they save you from reinventing common document workflows. Second, they demonstrate what a skill is supposed to feel like from the user's point of view: you describe the result you want, and Claude loads the specialist instructions it needs.

To see your installed skills:

1. In Claude's left sidebar, select **Customize**, and then select **Skills**.
2. Review the Anthropic skills and any personal, shared, or organization skills available to you.
3. In the **Skills** list, select **morning**.



4. Read the **morning** skill's description that is used by Claude to decide when to run it:

Render the user's morning brief as a styled HTML artifact, or set it up as a recurring weekday task. Use only when the user explicitly asks to run, see, or set up their morning brief, or if they invoke /morning by name. A question about their day, schedule, or calendar is not by itself a request for the brief; answer it directly instead.

5. Drop down the list of files distributed with the skill and note the **morning** skill includes a custom font named **Fraunces** with an **Open Font License (OFL)**.
6. Make sure the skill you want to use is enabled.
7. Select the skill's menu and note you can remove the skill or try it in chat.
8. You can toggle the display between formatted view and code view to see the raw Markdown that defines the skill:

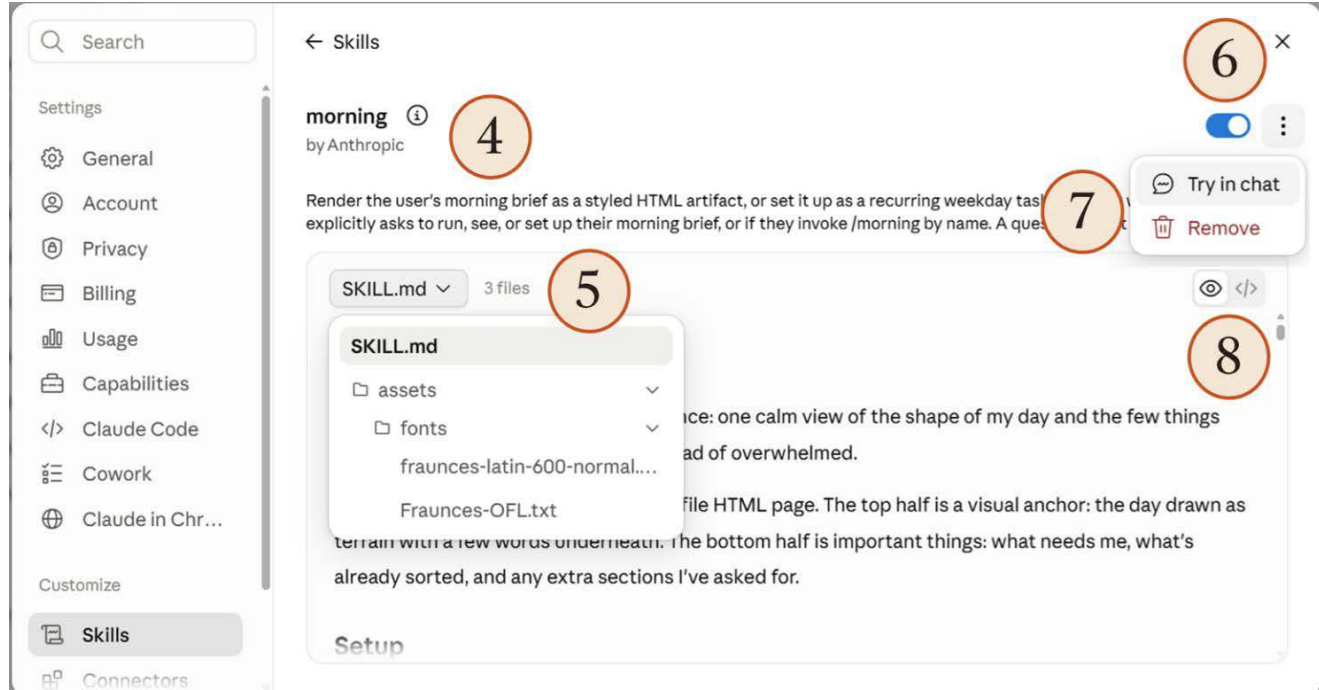


Figure 3.3C: Viewing the built-in morning skill by Anthropic



Watch out: I do not use the **morning** skill because it asks for full write access to my Google Calendar even though I have given it permission to read all my calendars. Write and delete permissions are separate and I don't want to give those permissions when it shouldn't be necessary. Luckily, skills are an open standard and Anthropic encourages people to look at their skills to learn how to customize your own.



Learn more online: *What are skills?* <https://support.claude.com/en/articles/12512176-what-are-skills>.

Browse and install a skill

You do not have to create every skill yourself. Claude has a directory where you can browse skills made available by Anthropic, partners, or your organization.

To install one:

1. If you're not already viewing skills, open **Customize | Skills**.
2. Select **Browse**.
3. Select **/brand-guidelines**, read its description, and then select **Install**.
4. Return to **Customize | Skills** and confirm that the skill is enabled.
5. Close **Directory** and **Settings**.
6. In a new chat, type **/** and note the slash commands available to you, including **brand-guidelines**:

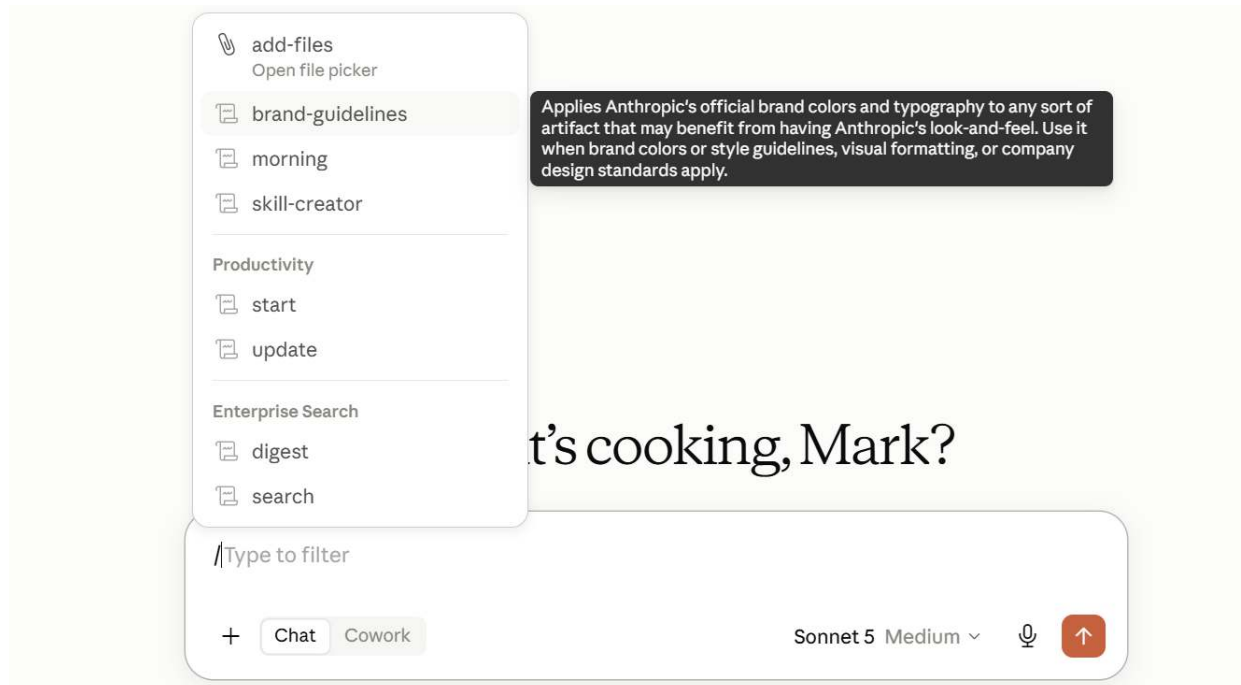


Figure 3.3D: Triggering a slash command

7. Select **brand-guidelines**.

8. Complete the rest of a prompt to try out the skill:

```
/brand-guidelines create a one-page Word document that explains the concept of Claude Skills with Claude branding.
```

9. Note the result uses Anthropic brand colors and font families:

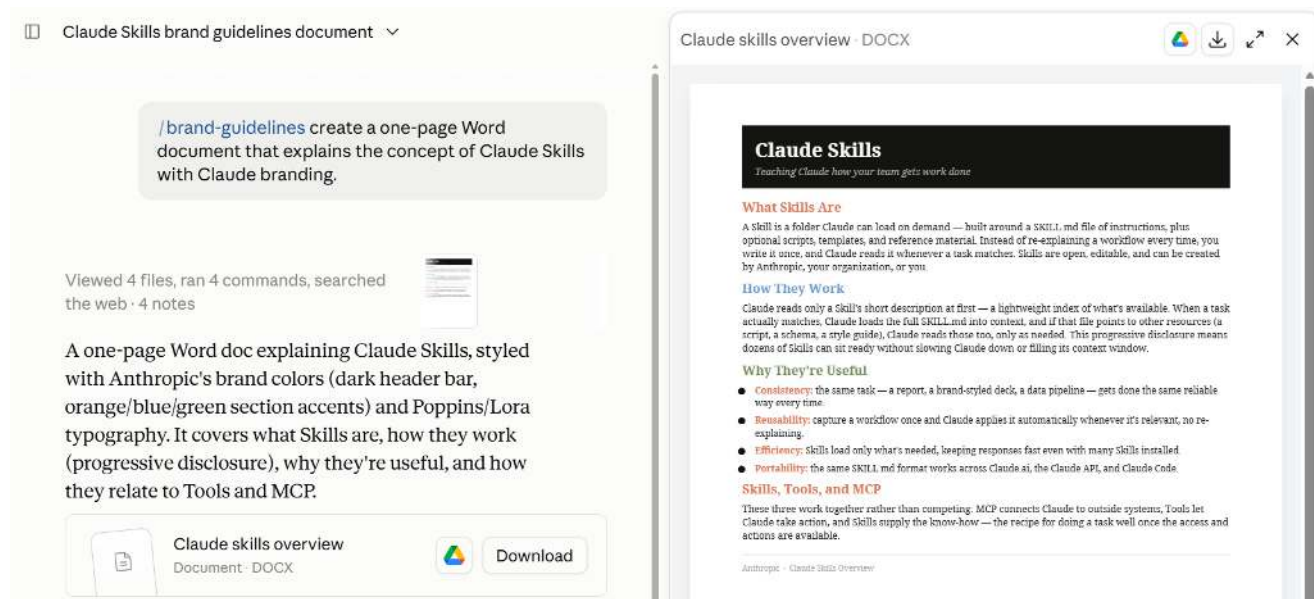


Figure 3.3E: Anthropic branding applied to a Word document about Claude Skills

An installed skill is view-only. You can use it, but you cannot directly change its contents. If you need different behavior, make your own skill rather than treating someone else's skill as if it were your private instructions.

On Team and Enterprise plans, the directory can also contain skills that your organization has approved and shared. A colleague can also share a skill directly with you. A directly shared skill appears in the **Shared with you** section and remains disabled until you turn it on.



Watch out: Only install skills from sources you trust. A skill can contain scripts, software dependencies, and instructions that cause Claude to access outside services. Read what a third-party skill contains before you enable it, especially if it includes executable code or asks Claude to connect to external systems.



Learn more online: *Use skills in Claude:* <https://support.claude.com/en/articles/12512180-use-skills-in-claude>.

Upload a custom skill that someone gives you

A custom skill can also arrive as a ZIP file. This is useful when a colleague, consultant, author, or training course gives you a skill directly rather than publishing it in a directory.

A skill ZIP contains a folder with at least a `skill.md` file. That file stores the skill's name, description, and instructions. It can also include reference files, templates, or scripts.

To upload one:

1. Save the ZIP file somewhere you can find it.
2. Open **Customize | Skills**.
3. Select the **Add** button, then **Upload a skill**.
4. Choose the ZIP file.
5. Review the new skill in your skills list.
6. Enable it only after you are satisfied that you trust its contents and understand what it does.

The skill you've just uploaded is private to your account unless you deliberately share it through a Team or Enterprise organization.



Good practice: Treat a skill ZIP like a small software package, not like an ordinary document attachment. Review its instructions, scripts, dependencies, and external connections before you install it.

You now know what skills are, why they are useful, which built-in skills Claude already provides, and how to install a skill made by someone else. The rest of this chapter focuses on the more valuable step: turning one of your own stable procedures into a skill that you can test and improve.

Select a stable repeated procedure

A **standard operating procedure (SOP)** is a documented, repeatable set of steps for carrying out a task consistently. In the context of *Book 3*, an SOP is essentially the human-readable procedure that you can later turn into a Claude skill or automation.

Choose a task you have done several times already. It should have a result you were satisfied with more than once. A stable procedure has steps that stay roughly the same each time, even if small details change. Avoid a procedure that is still evolving. If you changed your approach last week and expect to change it again, wait until it settles down before turning it into a skill.



Real-world example: A marketing coordinator writes a short social media summary of each blog post the team publishes. She has done this the same way for two months: read the post, pull out the main point, write three sentences, and add one call to action. That consistency makes it a strong candidate for a skill.

Define required inputs and outputs

Write down exactly what the skill needs to start, and exactly what it should produce when it finishes.

- **Inputs** might include a document, a topic, a target audience, or a word limit
- **Outputs** might include a document, a table, a short summary, or a checklist

Being specific here prevents a common problem: a skill that runs but produces something close to what you wanted instead of what you actually asked for:

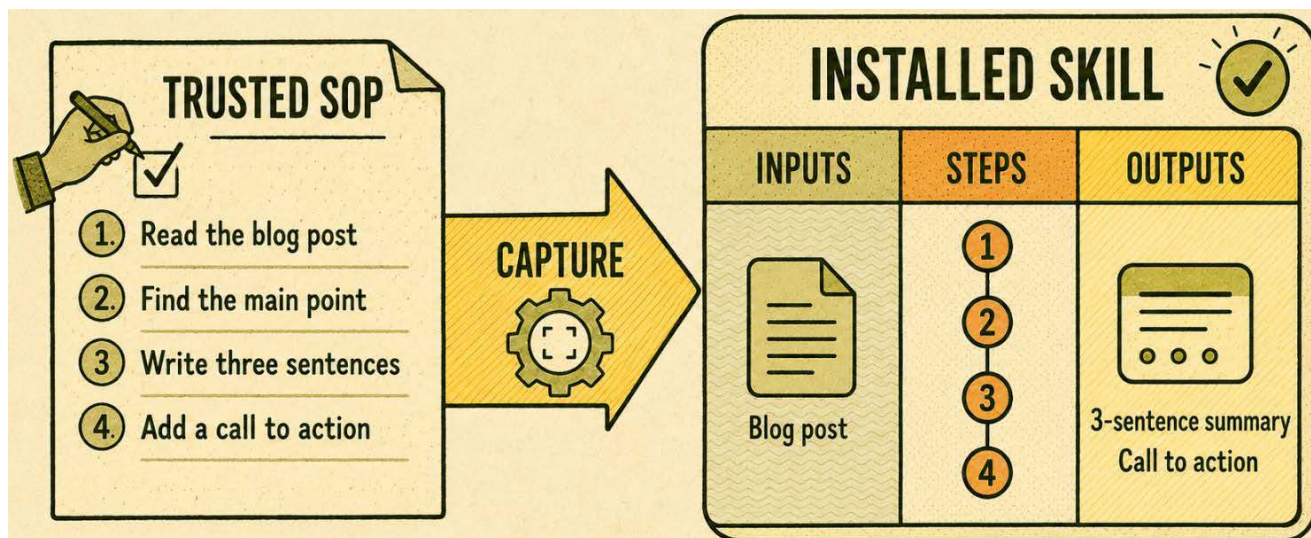


Figure 3.3F: A reusable skill starts as a SOP you have already tested by hand

Capture rules, resources, and examples

List the rules the procedure must follow, such as a required format, a length limit, a tone, or a fact that must always be checked. Then gather any resources the skill will need, such as a template, a style guide, or a reference document.

Add at least one worked example: a real input and the exact output you wanted from it. An example often communicates a rule faster than a written instruction can.

Write acceptance criteria

Acceptance criteria are the specific conditions a result must meet before you consider the skill's work finished. Write them as a short checklist, not a paragraph.

For the marketing summary example, acceptance criteria might include:

- Exactly three sentences, not four or two
- One clear call to action at the end
- No claim that is not stated in the original blog post



Sample files: A one-page Word specification for a Blog Post Social Summary skill: <https://github.com/markjprice/claude-vb/tree/main/Files/Book3/3-3-Blog-Post-Social-Summary-Skill-Specification.docx>. It includes numbered fields for name, inputs, steps, rules, worked example, and acceptance criteria.



Good practice: Write the acceptance criteria before you build the skill, not after. Deciding what “good” looks like in advance keeps you from lowering your standard without noticing once you see the first result.

You now have a clear specification: inputs, rules, an example, and acceptance criteria. Turning a procedure into instructions is only half the job.

The next sections turn that specification into a working skill, test it, and share it if it proves reliable.

Create the skill through conversation

Let's create a skill:

1. Open a new chat and tell Claude you want to create a skill using its skill-creator skill. (Yes, Claude has a predefined skill for creating more skills.)
2. Provide the inputs, rules, resources, and example from your specification.
3. Ask Claude to draft the skill's instructions and confirm the required inputs and outputs.
4. Review the draft against your acceptance criteria before saving it.

Use the skill specification from the previous section as an attachment, then enter this prompt:

Create a reusable skill using your skill-creator skill from the attached Blog Post Social Summary specification. Help me refine it through conversation before you create it. Ask one focused question at a time if any required detail is missing or unclear.

When you have enough information, show me a concise summary of the skill's purpose, inputs, steps, rules, example output, and acceptance checks. Wait for my approval, then create, validate, and install the skill. Keep it simple and do not add features that are not in the specification.

Claude should now summarize what it found in the Word document skill specification, but it might have questions that you need to answer like what should trigger the skill:

I've read the spec. It's a well-defined skill — here's what it covers:

- **Inputs:** blog post (pasted text / uploaded doc / link), target audience (required), optional CTA
- **Steps:** identify main point/audience/action → list supporting facts → draft exactly 3 sentences (CTA in the last) → fact-check against source, return summary only
- **Rules:** plain professional language, no hashtags/emoji unless asked, no invented facts, ask one question if a required input is missing
- **Example and acceptance criteria** are both given

One thing the spec doesn't cover: when Claude should actually reach for this skill. That description is what determines whether it triggers in conversation, so I want to get it right.

When should this skill trigger? ×

- ☒ 1 Whenever a blog post + request for a social summary/blurb appears 🔗
- ☐ 2 Only when the user explicitly says 'social summary' or names this skill
- ☐ 3 Broader: any request to summarize a blog post into a short promotional blurb (social post, email teaser, e...)
- ☐ Something else

Skip

Figure 3.3G: Claude asking for the trigger before creating a new skill



I could have included the trigger in the specification, but I wanted you to see what happens when you define a specification with some missing information.

Answer Claude's questions:

1. Select 2 so the skill only runs when explicitly named.
2. When Claude asks, "Want me to go ahead and create, validate, and install it?", enter Yes.
3. When the `blog-post-social-summary/SKILL.md` folder and file is created and shown in the **Artifacts** panel, select the **Code** button to see the raw Markdown file that defines the skill:

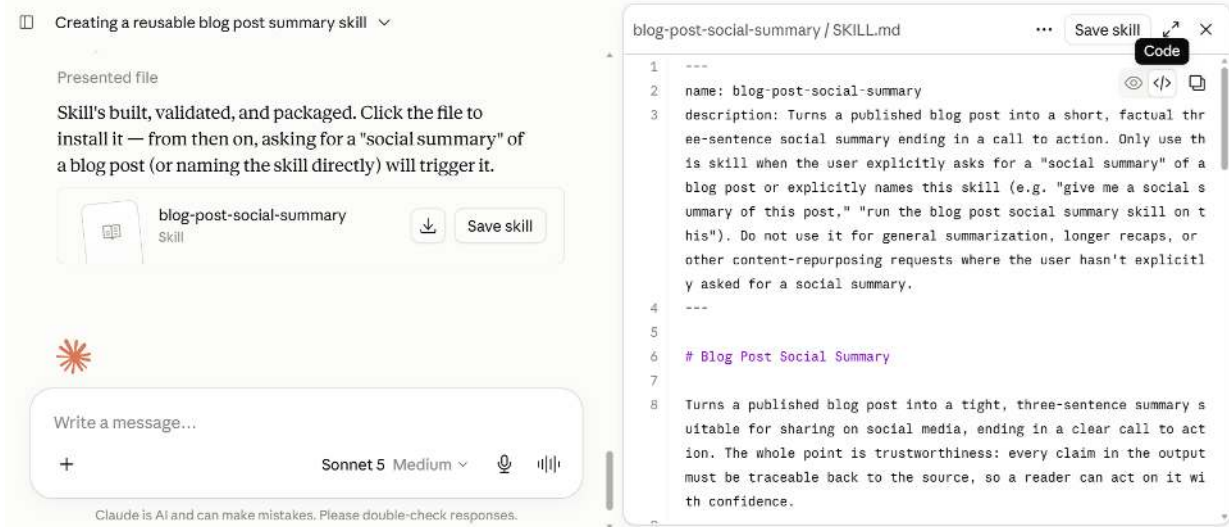


Figure 3.3H: The created skill showing its Markdown skill.md file

4. Select **Save skill**.
5. Close the file.
6. In the Claude sidebar, select **Customize** and then **Skills**, and confirm your new skill is listed and enabled with **You** as the **Author**.
7. Select your skill, review its details, and then select the menu and **Try in chat**:

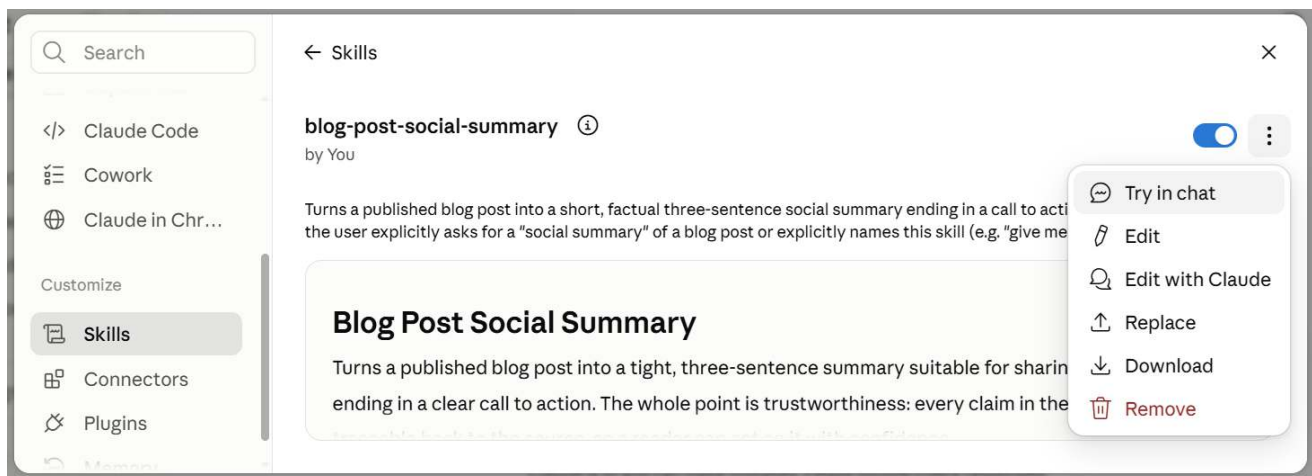


Figure 3.3I: Your skill and its action menu

Test normal and difficult cases

Run the skill on a normal example first, one like the worked example in your specification. Check the result against your acceptance criteria.

Test the installed Blog Post Social Summary skill with this prompt:

```
/blog-post-social-summary Test the Blog Post Social Summary skill with this normal example:
```

```
Blog post: ClearPath will offer free 30-minute online planning clinics every Thursday at 1:00 p.m. ET, starting September 10. Customers can bring one question about setting up their first project. Registration is required through the ClearPath website.
```

```
Audience: New ClearPath customers
```

```
First, run the skill and show its three-sentence social summary. Then check the result against every acceptance criterion in the skill specification. For each criterion, report Pass or Fail and briefly explain why. Do not change the result during the check. If anything fails, suggest a correction separately. Do not test a difficult case yet.
```

The response should show two distinct parts:

1. The result produced by the skill
2. An acceptance check confirming whether it contains exactly three sentences, ends with a clear call to action, and uses only facts traceable to the source

Acceptance criteria check:

Criterion	Result	Explanation
Output contains exactly three sentences	Pass	Three sentences, each ending in a period, no extras.
Final sentence contains one clear call to action	Pass	"Register on the ClearPath website" is the single, unambiguous CTA; the trailing clause ("bring one question...") is supporting detail from the source, not a competing action.
Every factual claim traces to the source	Pass	Free / 30-minute / online / "planning clinics" / every Thursday / 1:00 p.m. ET / starting September 10 / registration required via ClearPath website / bring one question about setting up first project — all stated directly in the source. Nothing invented (no hashtags, emoji, extra offers, or dates added).

All three criteria pass — no correction needed for this case.

Figure 3.3J: A test run of the skill and an acceptance check

Keeping the first test deliberately straightforward establishes that the skill works under ordinary conditions before the reader challenges it with missing, ambiguous, or potentially misleading inputs.

Then test at least one difficult case: incomplete information, an unusual request, or an input that breaks the expected pattern. A skill that only works on easy cases is not ready to trust.

Choose a difficult case that tests one of the skill's rules. For the Blog Post Social Summary skill, leave out the required target audience. The skill should notice the missing input and ask one focused question instead of guessing.

Enter this prompt:

```
/blog-post-social-summary Test the Blog Post Social Summary skill with this
incomplete input:
```

```
Blog post: ClearPath has published a new guide to planning a first project. The
guide explains how to set a goal, divide the work into smaller tasks, and choose
a realistic completion date. Readers can download the guide from the ClearPath
website.
```

```
I have not provided a target audience.
```

```
Run the skill once. If it asks a question, stop and show that question. If it
produces a summary, show the summary without changing it. Then report Pass or Fail
for each of these checks:
```

```
Did the skill avoid guessing the missing audience?
```

```
Did it ask one focused question?
```

```
Did it avoid inventing any facts?
```

```
Do not correct the result during the test.
```

A passing result should ask for the target audience and wait for your answer. It should not produce the three-sentence summary yet. If it guesses an audience or continues without one, the skill has failed one of its own rules:

Question:

Who is this summary for — e.g. new ClearPath customers, existing customers, or another audience?

Checks:

Check	Result	Explanation
Did the skill avoid guessing the missing audience?	Pass	No summary was drafted; the missing audience wasn't assumed or filled in.
Did it ask one focused question?	Pass	A single question was asked, about the one missing piece (audience), with nothing else bundled in.
Did it avoid inventing any facts?	Pass	No facts were stated at all, so nothing could be invented — no summary content was produced to check.

Figure 3.3K: The skill asks you a question and waits for your response

Reply to the question with:

```
New ClearPath customers who are planning their first project.
```

The skill should now produce its summary. Check the finished result against the original acceptance criteria: exactly three sentences, one clear call to action in the final sentence, and no facts that cannot be traced to the source:

```
New to ClearPath? Check out our new guide to planning your first project – learn how to set a goal, break the work into smaller tasks, and pick a realistic completion date. Download the guide from the ClearPath website.
```

If a test fails, do not keep correcting each result by hand. Return to the skill instructions and improve the rule that allowed the failure. Then repeat both the normal test and the difficult test. This checks that your change fixes the problem without breaking behavior that already worked.



Watch out: Do not judge a skill after one successful run. Test it three or four times with different inputs before you rely on it for real work.

Invoking the skill

Once the skill has passed its normal and difficult tests, you can use it on a real task. Invoke the skill by typing a slash and selecting or typing its name, and then provide the required inputs in the prompt.

For example, enter:

```
/blog-post-social-summary
```

```
Blog post: ClearPath will offer free 30-minute online planning clinics every Thursday at 1:00 p.m. ET, starting September 10. Customers can bring one question about setting up their first project. Registration is required through the ClearPath website.
```

```
Target audience: New ClearPath customers
```

The skill produces:

```
Starting September 10, ClearPath will offer free 30-minute online planning clinics every Thursday at 1:00 p.m. ET. New ClearPath customers can bring one question about setting up their first project. Register through the ClearPath website to attend.
```

The invocation tells Claude which skill to use. The blog post and target audience supply the inputs required by that skill. You do not need to repeat the skill's steps or rules because they are already stored in its instructions.

In the real world, you would normally invoke the skill with a public link. You can test it with a sample file for this book:

```
/blog-post-social-summary
```

```
Blog post:
```

```
https://raw.githubusercontent.com/markjprice/claude-vb/main/Files/Book3/3-3-blog-post.md
```

```
Target audience: New ClearPath customers who are planning their first project.
```

The sample public blog post announces free weekly planning clinics and includes dates, times, eligibility, capacity, registration details, and a clear next action. It also identifies ClearPath as fictional, preventing readers from mistaking the example for a genuine offer.

You can also invoke the skill and wait for it to ask for information it needs interactively:

1. Invoke the skill by typing a slash and its name:

```
/blog-post-social-summary
```

2. When Claude asks for the blog post, reply with the link:

```
https://raw.githubusercontent.com/markjprice/claude-vb/main/Files/Book3/3-3-  
blog-post.md
```

3. When Claude asks who this is for, reply with the audience:

```
New ClearPath customers who are planning their first project.
```

4. Claude should reply with a suggested social posting:

```
New to ClearPath? Free online planning clinics start September 10, running  
every Thursday at 1:00 p.m. ET for 30 minutes with a ClearPath project  
specialist. Email clinics@clearpath.example with your name and preferred  
date to request a place, since each session is limited to 20 customers and  
registration closes at noon ET the day before.
```

The complete interaction is shown in *Figure 3.3L*:

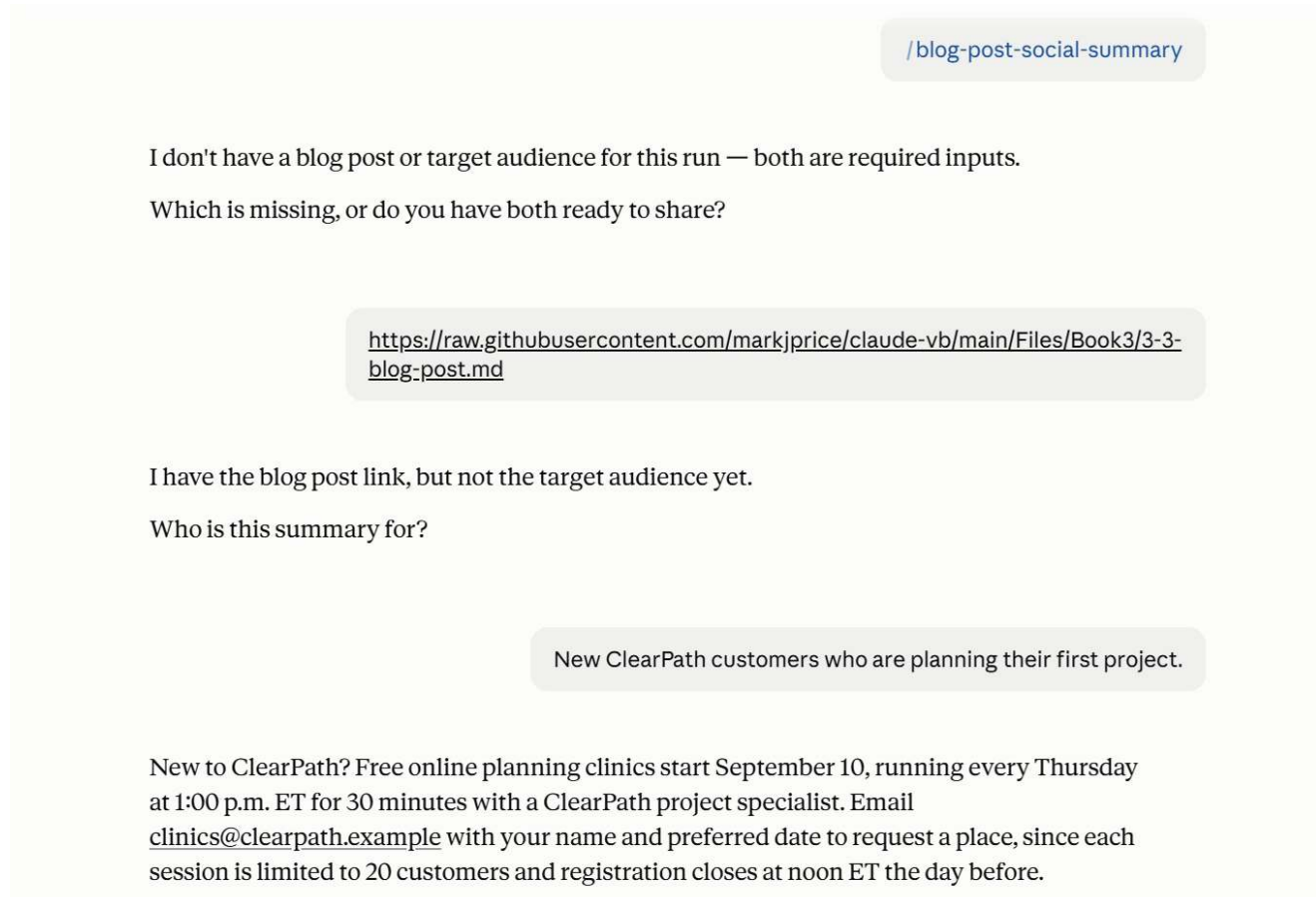


Figure 3.3L: Interactive conversation using the skill

Improve instructions and error handling

When a test fails, do not try again and hope for a better result. Read the failure, decide which instruction was missing or unclear, and update the skill's instructions directly.

Add error handling for cases where the skill cannot complete safely, such as missing information. Tell the skill to ask a clarifying question or stop and explain what it needs, rather than guessing.



Watch out: A skill is not magic. It is closer to a SOP that an unreliable worker is attempting to follow. The reason to create a reusable skill is not merely to save a prompt. It is to encode a known-good procedure. But consequential workflows still need validation that the procedure was actually followed. Reliable automation therefore benefits from: procedure → observe execution → check each important stage → detect deviations → verify completion.



Learn more online: *How to create custom skills:* <https://support.claude.com/en/articles/12512198-how-to-create-custom-skills>. *SkillSentry: Reliable Skill Execution for LLM Agents via Runtime Assurance:* <https://arxiv.org/html/2608.09253v1>.

Try it now

Turn the procedure you identified in *Chapter 3.1* into a skill. Write down its inputs, outputs, rules, one worked example, and three acceptance criteria. Keep this list. Then build the skill you specified. Run it once on a normal case and once on a difficult case. Revise its instructions at least once based on what you observe.

Check the result

- ☐ Have you done this procedure the same way at least twice before?
- ☐ Could someone else follow your written inputs, rules, and example without asking you a question?
- ☐ Do your acceptance criteria describe something you can check by looking at the result, rather than a vague feeling?
- ☐ Did the skill meet every acceptance criterion on the normal test?
- ☐ What happened on the difficult test, and did you update the instructions in response?
- ☐ Would you feel comfortable sharing this skill with a colleague today?

A skill handles one procedure well. *Chapter 3.4* shows you how to go further and build a library of skills, not just a single task.

Chapter 3.4 Build a Library of Skills for Real Work

Chapter 3.3 showed you how to turn one stable procedure into a skill, test it, and improve it. One skill can save you from repeating the same instructions. A small library of well-chosen skills can do something more useful: capture the reliable parts of how you work.

This chapter looks at practical skill ideas for eight common professions. You do not need to work in these jobs to use the examples. Look at the pattern behind each one. A lawyer reviews against rules. A marketer turns source material into audience-specific content. An HR professional compares information against fixed criteria. A real estate professional transforms property facts into consistent client-facing material. The same patterns appear in many kinds of work.

The goal is not to automate a profession. It is to identify small procedures inside a profession that can be described, checked, and reused.

Build skills around procedures, not job titles

“Make Claude a lawyer” is not a good skill specification. It is too broad, and it invites Claude to make judgments that should remain with a qualified person.

“Check a draft contract against our standard contract-review checklist” is much better. It has a defined input, a repeatable method, and a result that a person can review.

The same rule applies in every field. Prefer:

- “Review this against our checklist” instead of “Do my job for me”
- “Prepare a first draft from approved facts” instead of “Decide what we should say”
- “Flag missing information” instead of “Guess what is missing”
- “Prepare questions for a person to answer” instead of “Make the final hiring decision”



Good practice: A skill should remove repeated setup work while leaving professional judgment with the person who owns the outcome.

Each profession example below is presented in the same table format, so you can compare skill ideas at a glance and reuse the structure for your own skill specifications. Every table breaks the skill down into the same handful of fields:

- **Skill name:** A short, descriptive label for the skill, used when you refer to it or trigger it by name
- **Purpose:** One sentence stating what the skill is for and who it helps
- **Inputs:** The information the skill needs before it can start, such as a document, a checklist, or a set of facts
- **Procedure:** The ordered steps the skill follows, written the way you’d explain the task to a new colleague
- **Output:** What the skill produces when it finishes, such as a report, a table, or a draft
- **Acceptance criteria:** Where the risk of error is higher, some tables add a short checklist of conditions the result must meet before you’d consider it finished

Reading the tables this way, rather than as isolated examples, makes the underlying pattern easier to see: define the inputs and the procedure precisely, and the output becomes predictable and easy to check.

There are eight example tasks for different professions:

- Contract Issue Spotter for a Lawyer
- Campaign Brief Builder for a Marketer
- Interview Question Pack for HR

- Listing Draft from Verified Facts for a Real Estate Agent
- Account Meeting Brief for a Salesperson
- Lesson Adaptation for a Teacher or Trainer
- Manuscript Consistency Check for a Publisher or Author
- Weekly Decision and Action Review for Operations

They are available online in the GitHub repository at the following link:

<https://github.com/markjprice/claude-vb/blob/main/docs/ch3-4-skills.md>

Once this procedure is reliable, *Chapter 3.2's* scheduling can run it regularly, and *Chapter 3.5's* connectors can gather information from approved systems.

Figure 3.4A shows how the building blocks combine:

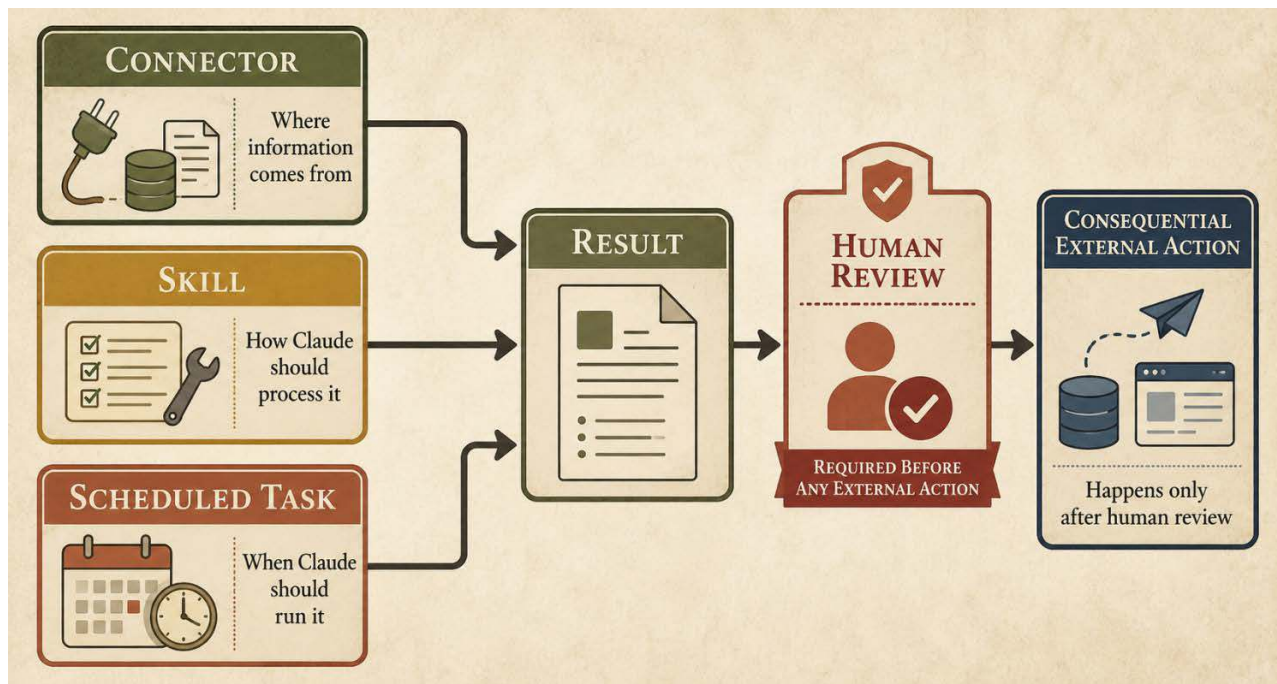


Figure 3.4A: How connectors, skills, and scheduled tasks mix together

Combine small skills instead of building one giant skill

You may notice that many examples can work together. A marketing workflow might use the following skills:

1. Research Evidence Summary
2. Campaign Brief Builder
3. Channel Adaptation
4. Brand and Claims Check

A hiring workflow might use the following skills:

1. Job Description Check
2. Interview Question Pack
3. Candidate Evidence Summary
4. Interview Notes Formatter

Claude can use more than one relevant skill during a task. The skills do not need to call each other directly. Claude can select the skills that match the work. This is another reason to keep them small. If each skill has a clear description, Claude has a better chance of choosing the right combination.



Good practice: Use the same acceptance-criteria approach from *Chapter 3.3* for every new skill. A useful skill library is not a folder full of prompts. It is a collection of procedures that have passed tests.

Know what should not become a skill

Not every repeated activity deserves automation. Do not turn a decision into a skill simply because you make it often.

Be especially careful when the task:

- Makes a legal, medical, financial, employment, or safety decision for another person
- Sends or publishes information without review
- Uses information you do not have permission to process
- Requires facts that change rapidly but has no trusted source for them
- Has exceptions that are more common than the standard procedure
- Produces consequences that are difficult to reverse

You can often automate the preparation without automating the decision.

For example, an HR skill can prepare a consistent evidence summary, but a person decides whether to advance a candidate. A legal skill can identify contract clauses, but a lawyer gives advice. A real estate skill can draft a listing, but an agent verifies the facts before publication.

Share a skill when the procedure belongs to the team

A skill becomes more valuable when several people need to follow the same procedure. Claude supports skill sharing on Team and Enterprise plans, provided the organization's owner has enabled the relevant sharing controls. You can share a skill with a specific colleague, a group that your organization has created, or your entire organization.

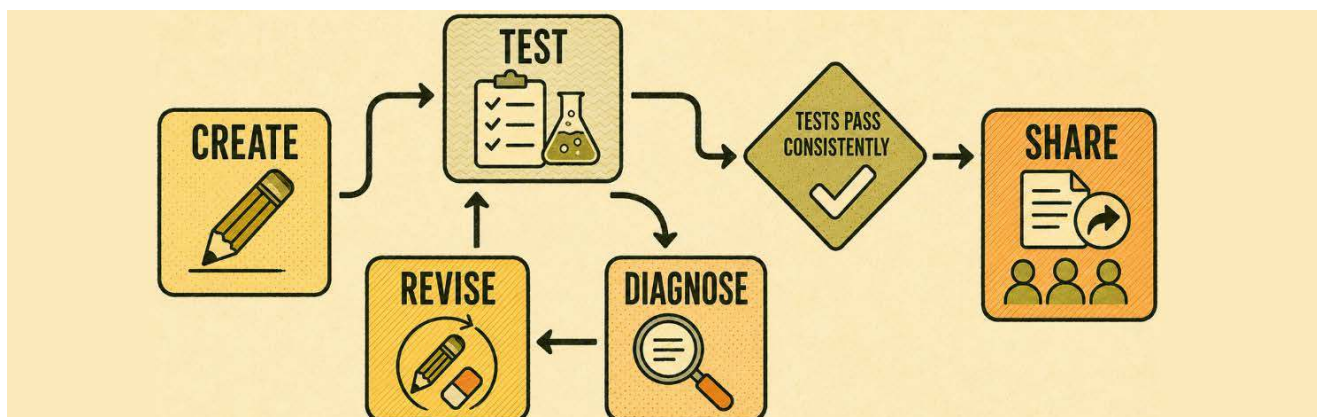


Figure 3.4B: Building a reliable skill means moving through create, test, diagnose, and revise, sometimes more than once



Watch out: If you are not on a Team or Enterprise plan, you are not able to use the share feature. Instead, you can download a skill as a ZIP file and manually share that file.

To share a skill you created:

1. Open **Customize | Skills**.
2. Open the skill.
3. Select **Share**.
4. Choose the people, group, or organization that should receive it.
5. Confirm the share.

A skill shared directly with a colleague or group appears in their skills list but is not enabled automatically. They choose whether to enable it. A skill shared with the whole organization can appear in the organization's directory for people to install.

Shared skills are view-only for recipients. They can use the skill but cannot edit your original. If you update the shared skill later, recipients receive the updated version automatically.

That last behavior makes sharing more powerful than emailing somebody a prompt. With a copied prompt, every person soon has a slightly different version. With a shared skill, the owner can improve one controlled version:

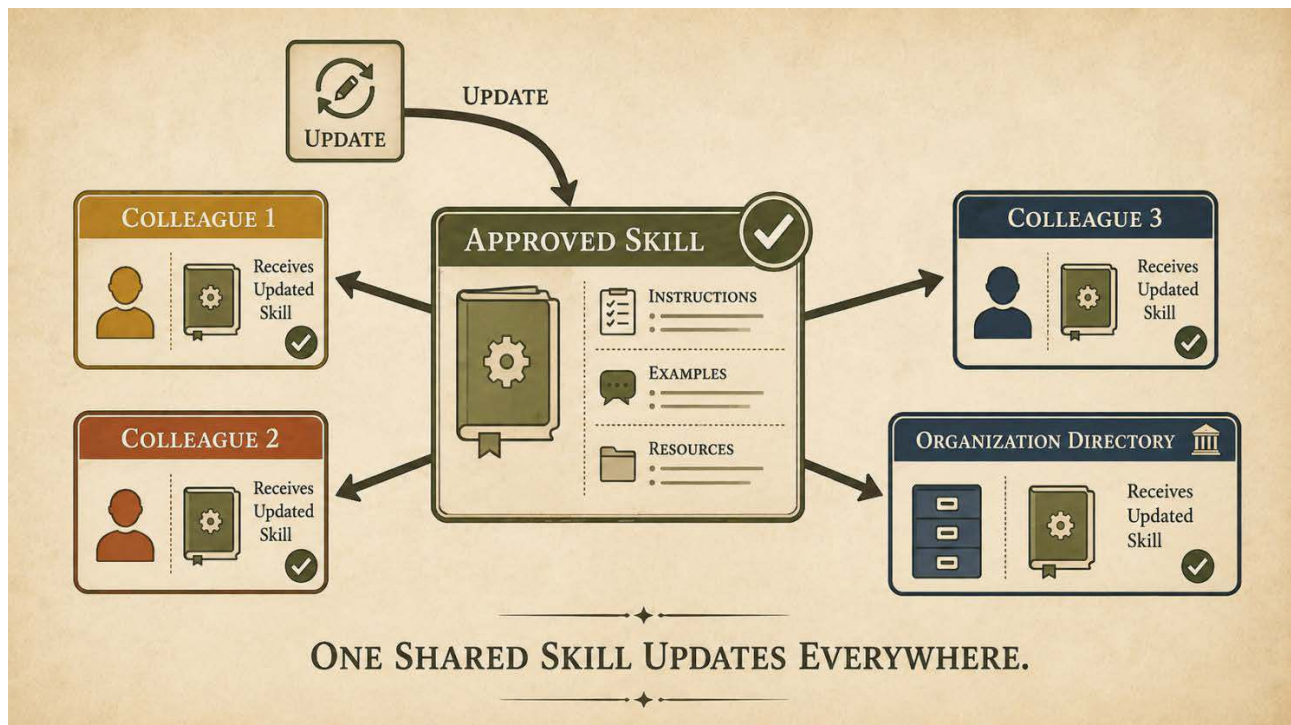


Figure 3.4C: Sharing maintains one reusable procedure instead of creating uncontrolled prompt copies



Watch out: Sharing does not make a skill trustworthy merely because a colleague created it. Review skills before enabling them, especially when they contain scripts, dependencies, or external connections.

For a larger organization, owners can provision approved skills for everyone. They can also bundle skills into plugins and make them available to specific groups. This is useful when, for example, the marketing team needs ten marketing skills but the rest of the organization does not.

Once a skill passes your tests consistently, you can keep it for your own use or share it. In an organization-managed workspace, a shared skill becomes available to colleagues who have permission to use it.

1. Open the skill's settings.
2. Choose to keep it private or share it with your workspace.
3. If sharing, write a short description so colleagues understand what the skill does and what it needs.
4. Confirm the sharing setting.

Anthropic's documented portable format is a skill folder containing `skill.md`, packaged as a ZIP for upload.



Good practice: Before sharing a skill, ask a colleague unfamiliar with the task to run it once. If they can use it correctly from your description alone, it is ready to share more widely.



Learn more online: *Provision and manage skills for your organization:* <https://support.claude.com/en/articles/13119606-provision-and-manage-skills-for-your-organization>.

Try it now

Choose one profession from this chapter that is closest to your own work, or choose another profession you know well:

1. Identify three repeated procedures in that job. For each one, write:
 - Skill name
 - One-sentence purpose
 - Required inputs
 - Five to seven steps
 - Expected output
 - Three acceptance criteria
 - One decision that must remain with a person
2. Build only the smallest of the three. Test it as you did in *Chapter 3.3*.
3. If you work in a Team or Enterprise organization and skill sharing is enabled, share the tested skill with one colleague and ask them to use it from the description alone.

Check the result

- ☐ Is the skill based on a narrow procedure rather than a broad job title?
- ☐ Could another person provide the inputs without needing you to explain the task?
- ☐ Does the skill avoid inventing missing information?

- ☐ Is there a clear point where professional judgment remains with a person?
- ☐ Does the output have acceptance criteria you can check?
- ☐ If you shared it, could the other person use it without copying your original prompt?

You now have more than a single reusable skill. You have a pattern for building a small library of tested procedures for real work. *Chapter 3.5* adds the next piece: trusted connections that let those procedures reach current information and tools outside Claude.

Chapter 3.5 Connect Trusted Information and Tools with MCP

Model Context Protocol (MCP) is a general standard for connecting Claude to information and tools, including ones built inside your own organization.

Why a standard connection protocol is useful

Before a shared standard existed, every connection between an AI assistant and an outside system needed its own custom work. MCP gives developers one common way to expose their information and actions, so Claude and other AI assistants can connect to many different systems without a separate integration for each one.

You do not need to understand how MCP works behind the scenes. You need to understand what it lets you connect, and how to check that a connection is trustworthy before you use it.

Several terms come up often when working with MCP:

1. **MCP host:** The AI application you are using, such as Claude.
2. **MCP client:** The connection inside that application that communicates with one MCP server.
3. **MCP server:** The outside service that exposes information and tools through MCP.
4. **MCP tool:** An action the server offers, such as searching, reading, creating, or updating information.
5. **MCP resource:** Information the server can provide.

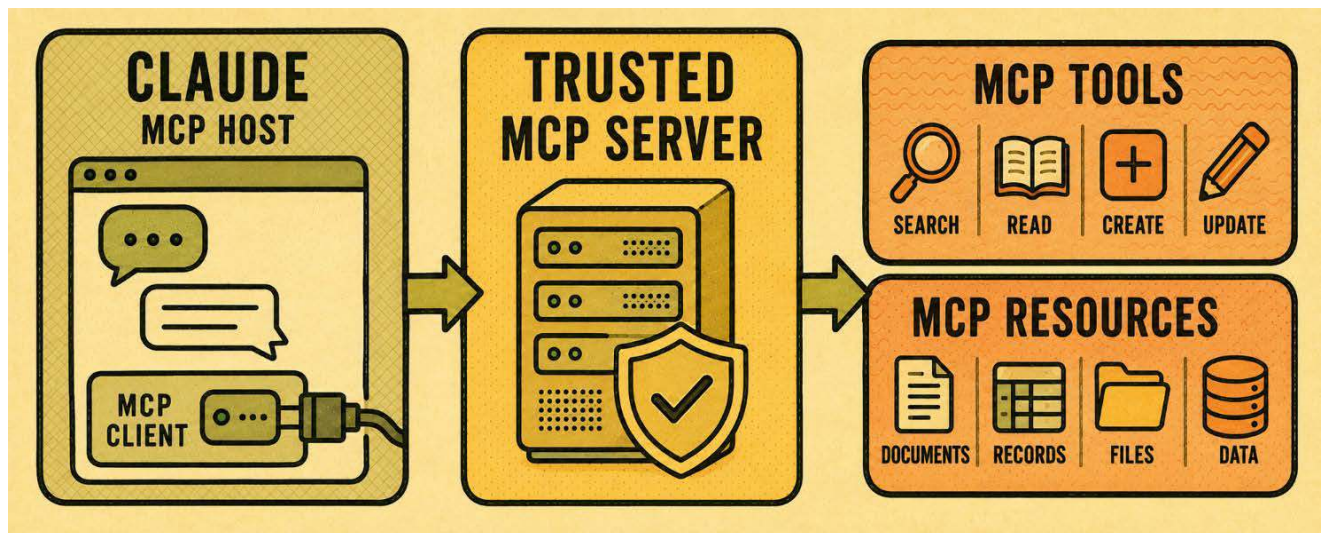


Figure 3.5A: Claude acts as an MCP client, communicating with a trusted MCP server that reaches an external system containing tools and resources on your behalf

An MCP server can run locally, on your own computer, or remotely, hosted by another organization. A local server is often used for personal tools and files. A remote server is common for company systems and third-party services.

For Claude Chat and Cowork, the easiest is a remote custom connector. Claude connects to that remote MCP server through Anthropic's cloud. Local MCP servers are a separate Claude desktop app mechanism and are not the same as adding a remote custom connector.

Remote servers usually need an account and explicit approval before Claude can use them. Treat this approval with the same care you would give any new connection to sensitive information.

Authentication, permissions, and trust

You should only connect to an MCP server you or your organization trusts. A server can request wide-ranging access, so review what it can see and do before approving it, just as you would with a plugin.

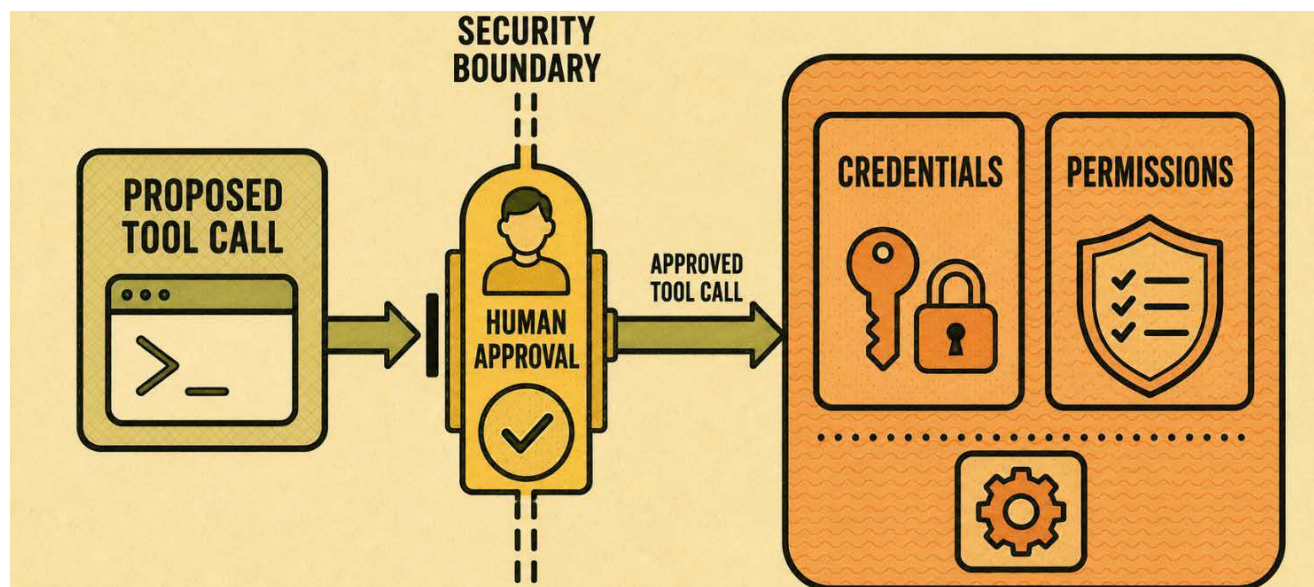


Figure 3.5B: Credentials and permissions sit inside a security boundary. A tool call should not cross it without your approval



Watch out: An unfamiliar or unverified MCP server can expose your data to a system you do not control. If your organization has an approved list of MCP servers, use only those. If it does not, ask before connecting one to work data.

Choose an MCP server built for the specific system you need, offered by the organization that owns that system whenever possible. Avoid a server from an unknown source, even if it claims to connect to the same system.

Notion's MCP server

Notion is an online workspace for documents, notes, databases, projects, and team knowledge. Its built-in AI can search that information, research across connected sources, summarize meetings, create and update content, and perform multi-step work through AI agents. Its main advantage is that the AI works inside the same workspace where your information and ongoing work already live.

A good example MCP server for this chapter's demonstrations is Notion's MCP server: <https://developers.notion.com/guides/mcp/overview>. It is well suited to a book example because:

- It is an official remote MCP server hosted by Notion
- It uses OAuth, so you will see a real permission flow
- It demonstrates both resource access and tools
- It is relevant to ordinary knowledge work

You will create a new and disposable Notion workspace for the demonstration, so you do not expose real work data while learning. It will demonstrate read tools, write tools, OAuth, and approval boundaries in one understandable example.



Learn more online: Sign up or login to Notion: <https://app.notion.com/signup>. Notion has a free plan and an optional desktop app but you can just use the web app for this chapter.

Create a Notion project page called **ClearPath Launch Notes** containing three fictional task items:

1. Draft landing page due September 4.
2. Customer webinar planned for September 10.
3. Final pricing still needs approval:

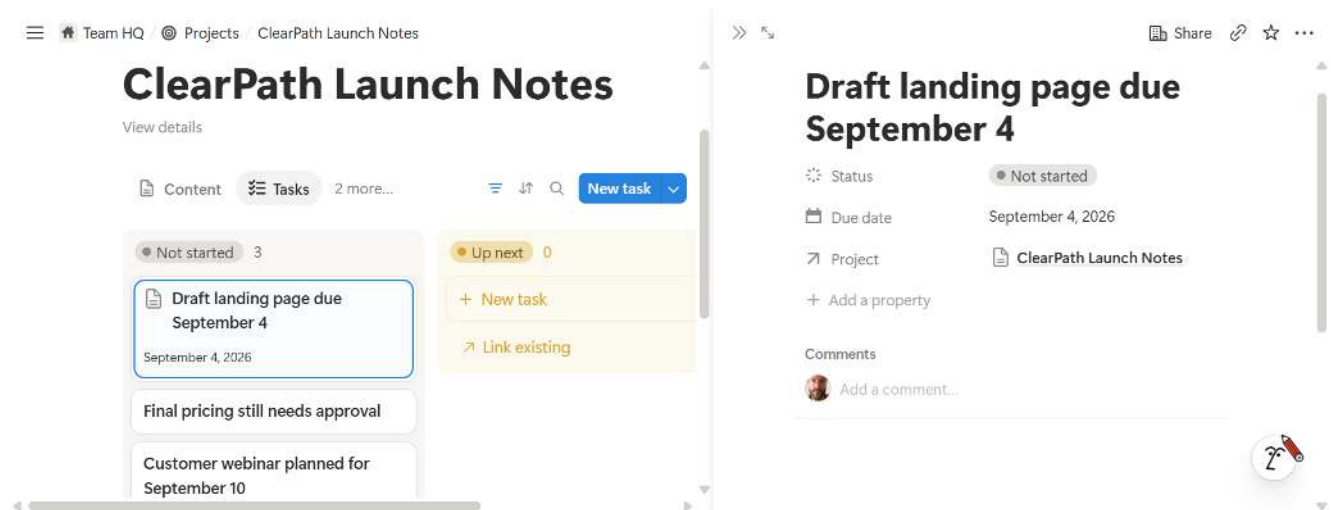


Figure 3.5C: A project in Notion with three tasks

Connecting the Notion MCP server

Next, you will connect Claude to the Notion MCP server.



Watch out: Use Claude on the web or the Claude desktop app to configure the MCP connection, then use the connection on the devices where your plan and current Claude app expose it. Check Anthropic's connector documentation because mobile setup capabilities are changing quickly.

To connect Claude to the Notion MCP server:

1. In the Claude sidebar, select **Customize**, and then select **Connectors**.
2. Select **Add**, and then select **Add custom connector** (or select **Browse connectors** and then search for Notion):

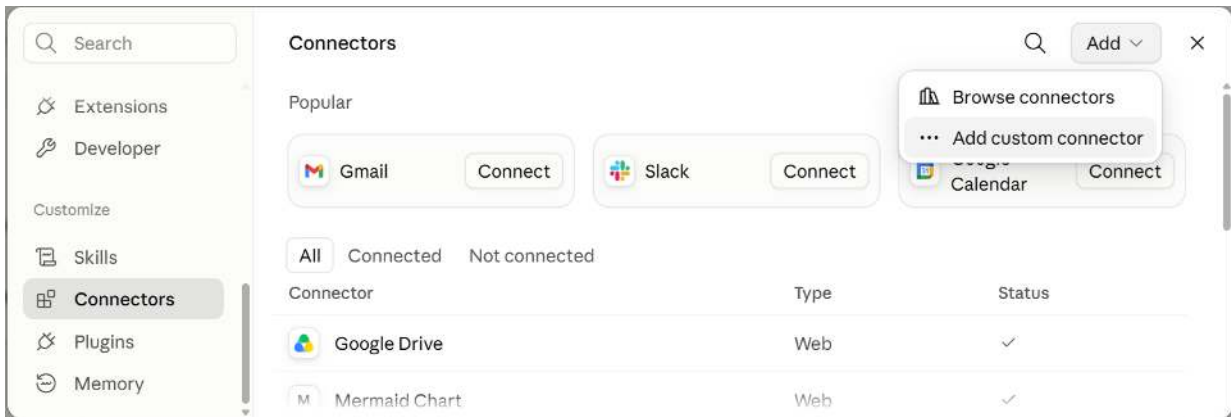


Figure 3.5D: Adding a custom connector

3. Enter a name like Notion MCP and the remote MCP server URL, `https://mcp.notion.com/mcp`:

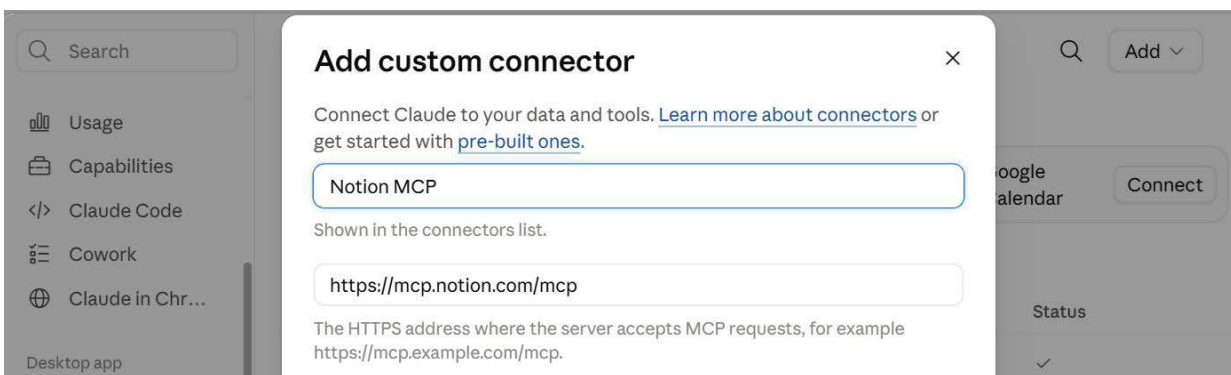


Figure 3.5E: Configuring the MCP connector name and URL

4. Select the **Add** button. Add advanced OAuth client settings only if the server requires them.
5. Select the **Connect** button and then complete OAuth when prompted:

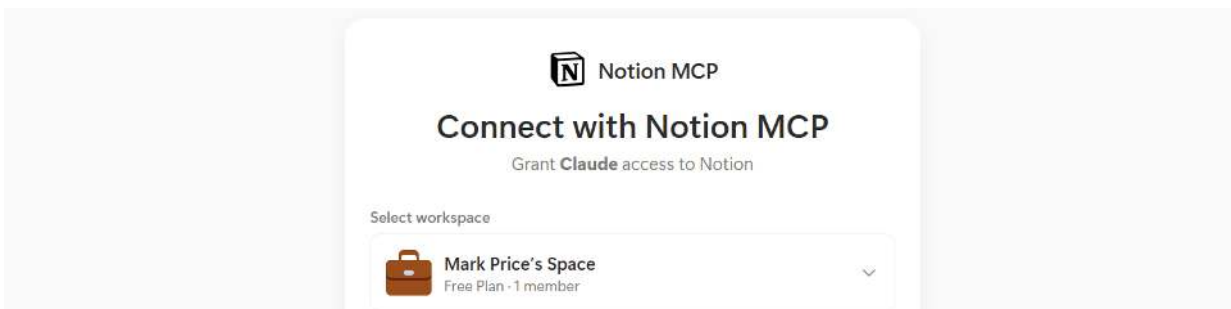


Figure 3.5F: Authorizing the MCP connection to Notion



Watch out: If something goes wrong, and you find yourself not able to add a connector because its URL is already registered on Claude's servers, but the MCP connector is not listed in your Customize page, try logging out of Claude and back in again. This often fixes the issue.

6. In the list of connectors, select the Notion MCP connector and review its tool permissions. They are grouped into two: **Read-only tools** and **Write/delete tools**. By default, the read-only tools should be set to **Always allow** since they are non-destructive. The write/delete tools should be set to **Needs approval**:

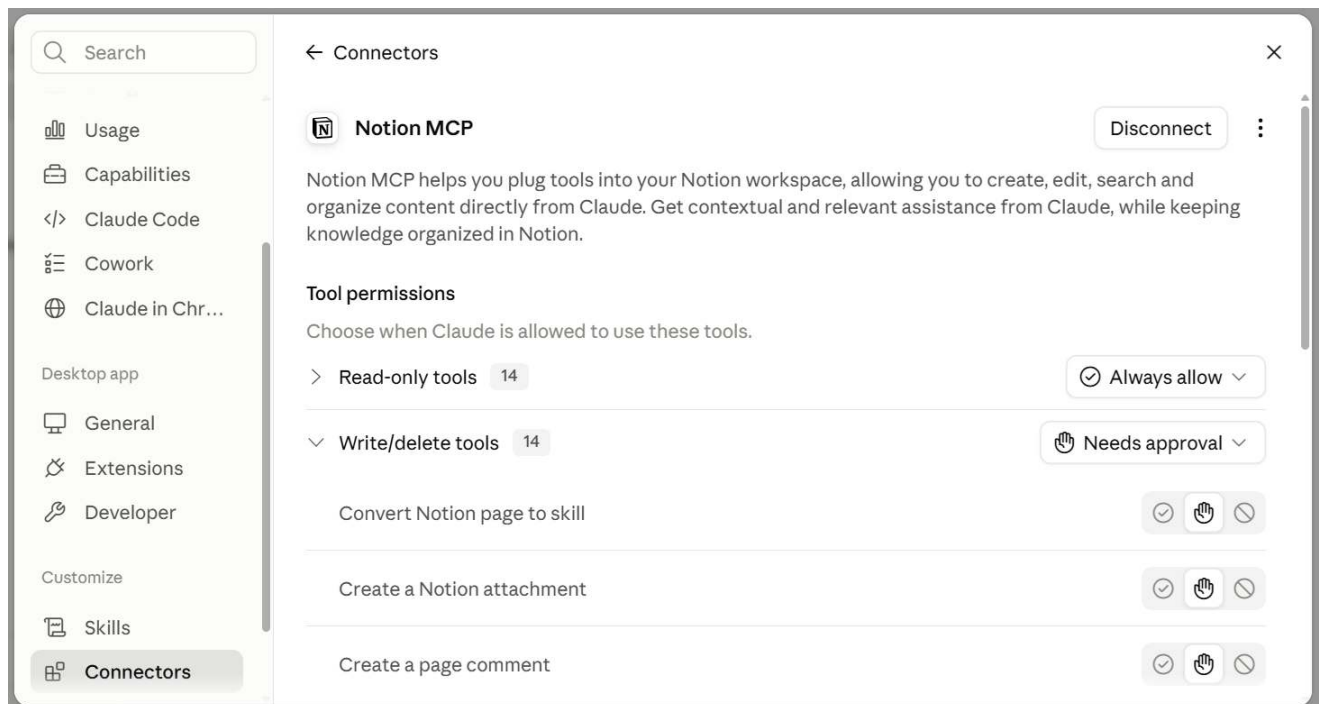


Figure 3.5G: Reviewing permissions for the Notion MCP connector



Watch out: For Team and Enterprise, an owner first adds the custom connector at the organization level, then individual members authenticate their own accounts. For Notion, use the official remote endpoint documented by Notion.

Trying out the Notion MCP connector

Let's try out the connector:

1. Start a new Claude chat conversation.

2. Select the + button and then confirm the Notion MCP connector is enabled:

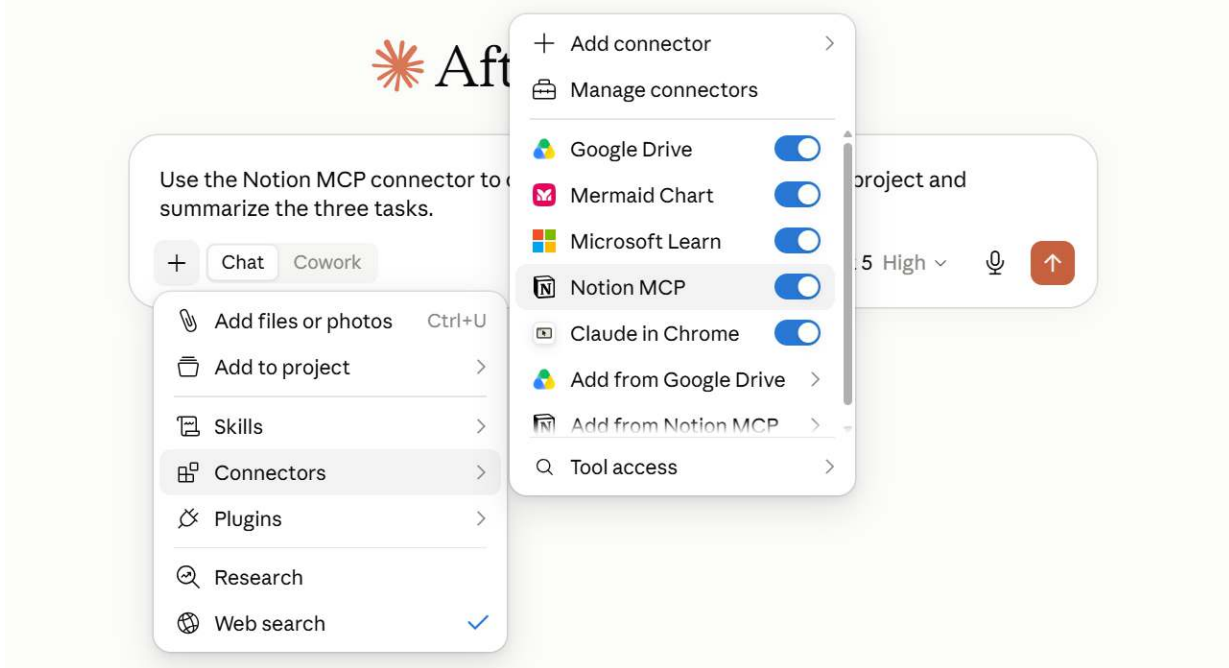


Figure 3.5H: Checking the status of the Notion MCP connector in a chat conversation

3. Send the following prompt and note the response:

Use the Notion MCP connector to open the ClearPath Launch Notes project and summarize the three tasks.

Used Notion MCP integration, loaded tools

All three tasks are still "Not started" — none have any content in the page body beyond the title/status.

1. **Draft landing page** — due September 4, 2026
2. **Customer webinar** — planned for September 10, 2026, no due date set on the task record itself
3. **Final pricing** — still needs approval, no due date set yet

The pricing approval is the one without a target date, so it's worth confirming whether it's blocking the landing page or webinar timelines.

Figure 3.5I: Response from Claude that used the Notion MCP connector to read tasks in your Notion project

4. Enter a prompt that will need to ask for permission before creating a new page called ClearPath Launch Summary:

Update the Draft landing page task to show a status of In progress, add a due date of 8 September for the webinar task, and create a new task named Create Webinar debrief slide deck planned for September 12.

- When asked for permission for each of the three individual updates, select **Allow once** for each one, and note the summary of actions taken:

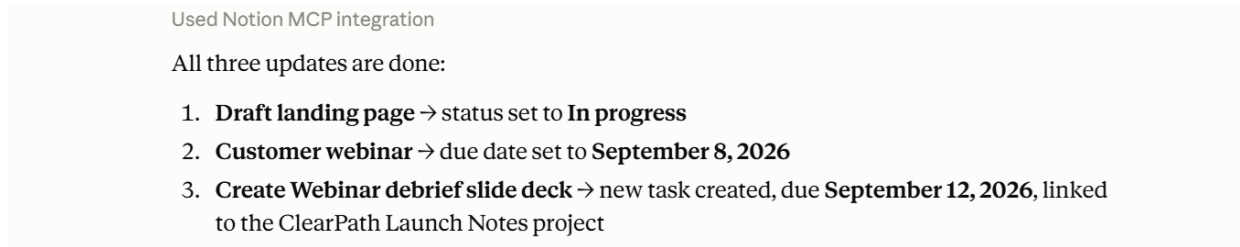


Figure 3.5J: Summary of the three updates that required permissions

- Return to Notion and check that the changes have been made properly:

ClearPath Launch Notes

View details

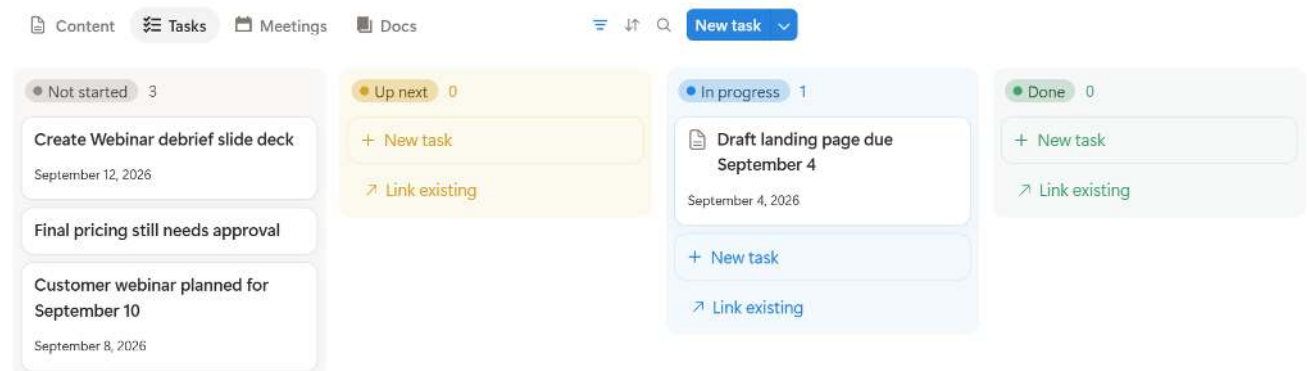


Figure 3.5K: Successful changes in Notion, including an In progress task

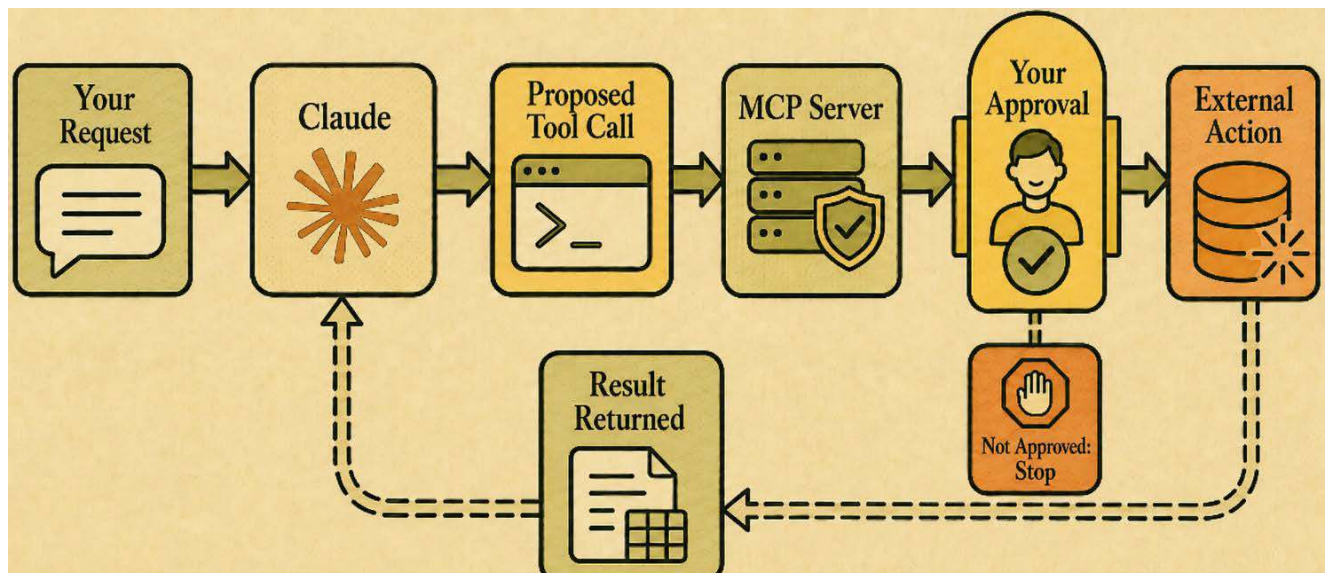


Figure 3.5L: A request becomes a tool call, an external action, and a result, with your approval sitting between the action and anything consequential

Test safely and handle failures

Test a new MCP connection with a low-risk request first, such as reading information rather than changing it. If the server fails or returns an error, do not repeat the request blindly. Check the error message and confirm the server is still trusted and available before trying again.



Good practice: Keep a short record of which MCP servers you have connected and why. This makes it easy to review your connections later and remove any that are no longer needed.

Try it now

If your organization has an approved MCP server, connect it and ask Claude to list its available tools and resources. If not, read the settings screen where a connection would be added, and note what information it would ask you to review.

Check the result

- ☐ Can you name the server you connected, or would connect, and who built it?
- ☐ Did you review the tools and resources it offers before treating it as trustworthy?
- ☐ Do you know what to do if a request to the server fails?

A trusted connector gives Claude a direct, structured way to reach information and tools in another system. When that route exists, it is usually the most reliable choice. But not every application, website, or older business system offers a connector or MCP server. *Chapter 3.6* covers the fallback: computer use, where Claude works through the same visible interface you do by opening apps, clicking controls, and typing on your behalf.

Chapter 3.6 Control Software with Computer Use

Chapter 3.5 showed how MCP connectors give Claude a direct, structured way to work with an outside service. A connector is usually the best option when one exists. But many programs, internal tools, and older systems do not have a connector.

Computer use gives Claude another route. Instead of calling a service through a connector, Claude can look at your screen and interact with software much as you do: point, click, type, open an app, and navigate through its interface.

This is powerful because it works with software that was never designed for AI. It is also less precise than a connector, so this chapter focuses as much on boundaries and supervision as on what computer use can do.



Watch out: At the time of writing in August 2026, computer use is a research preview for Claude Pro and Max users. It is available through Cowork and Claude Code in the Claude desktop app for macOS and Windows. Team and Enterprise users do not currently have this computer-use preview. Check Anthropic's current help page before publication because availability can change quickly.

Understand where computer use fits

When Claude needs to interact with another system in Cowork, it should use the most precise method available. The order is recommended:

1. **Connector:** Use a direct connection such as Gmail, Google Drive, Slack, or another supported service.
2. **Browser:** If there is no suitable connector, use Claude in Chrome to navigate a website.
3. **Screen interaction:** If the work must happen in a desktop app or another interface without a better tool, use computer use to click and type on the screen.

It is important to follow this order. A connector sends structured requests directly to a service in a controlled manner as you saw in *Chapter 3.5*. Screen interaction with computer use has to interpret pixels, buttons, menus, and changing layouts. It is slower and more likely to make a navigation mistake:

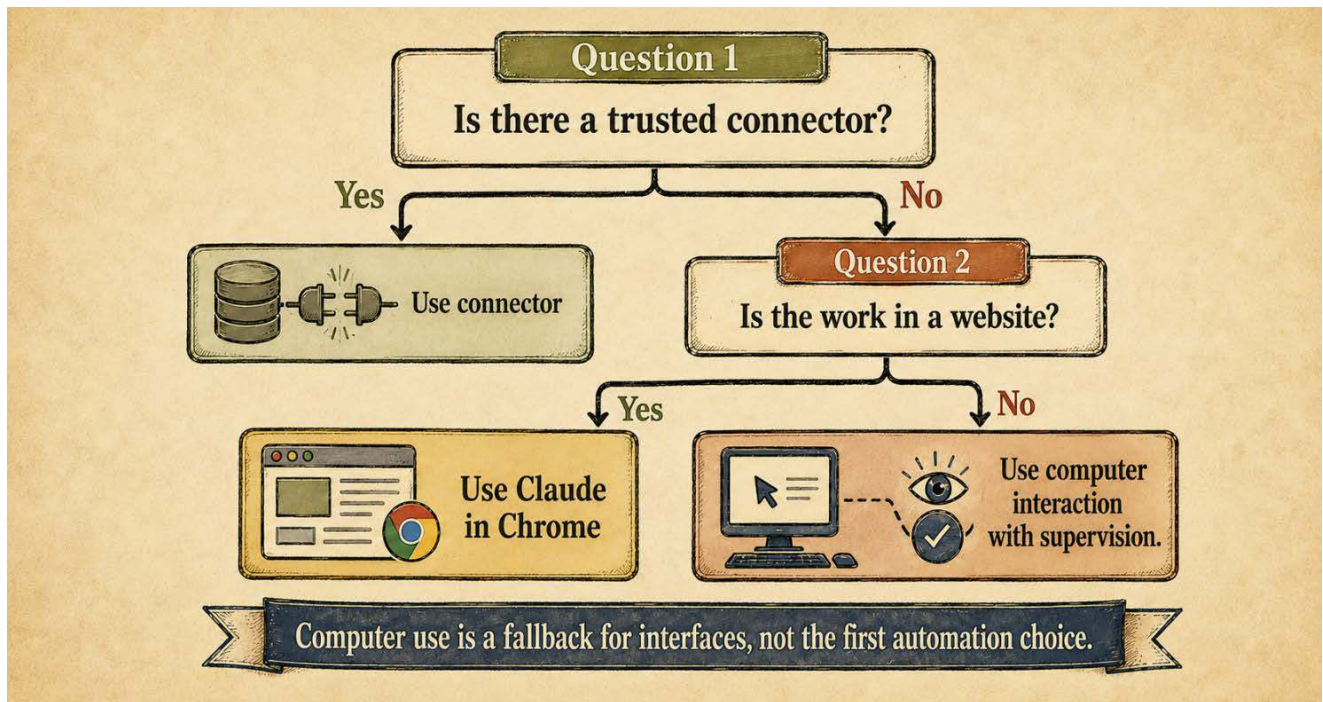


Figure 3.6A: Decision diagram showing that computer use is a fallback for interfaces, not the first automation choice.



Good practice: Do not use computer use merely because watching Claude move the pointer looks impressive. Use the most reliable tool that can complete the task.

Know what computer use can do

With permission, Claude can interact with apps and files on your computer. Anthropic gives examples such as opening a locally installed app that has no connector, filling a spreadsheet from several sources, working through an internal dashboard, opening a phone simulator and interacting with an app under development, and navigating a browser when no direct integration is available.

Computer use is different from Claude simply editing a file. If Claude has a direct file tool for the task, it may be able to change a document without opening the program that normally edits it. Computer use becomes useful when the interface itself is part of the task.

For example, imagine that your company has an old desktop program for logging equipment checks. There is no API, no connector, and no export command. A person normally opens the program, chooses a site, enters the date and readings, and saves the record. Computer use can potentially follow those visible steps.

That does not mean you should immediately let it update hundreds of records. Start with one test record that is easy to undo.

Turn on computer use

Before Claude can control desktop software, you must enable the feature and approve the apps it is allowed to use:

1. In the Claude desktop app, open **Settings**.
2. In the **Desktop app** section, select **General**.
3. Scroll down to **Computer use** and turn it on:

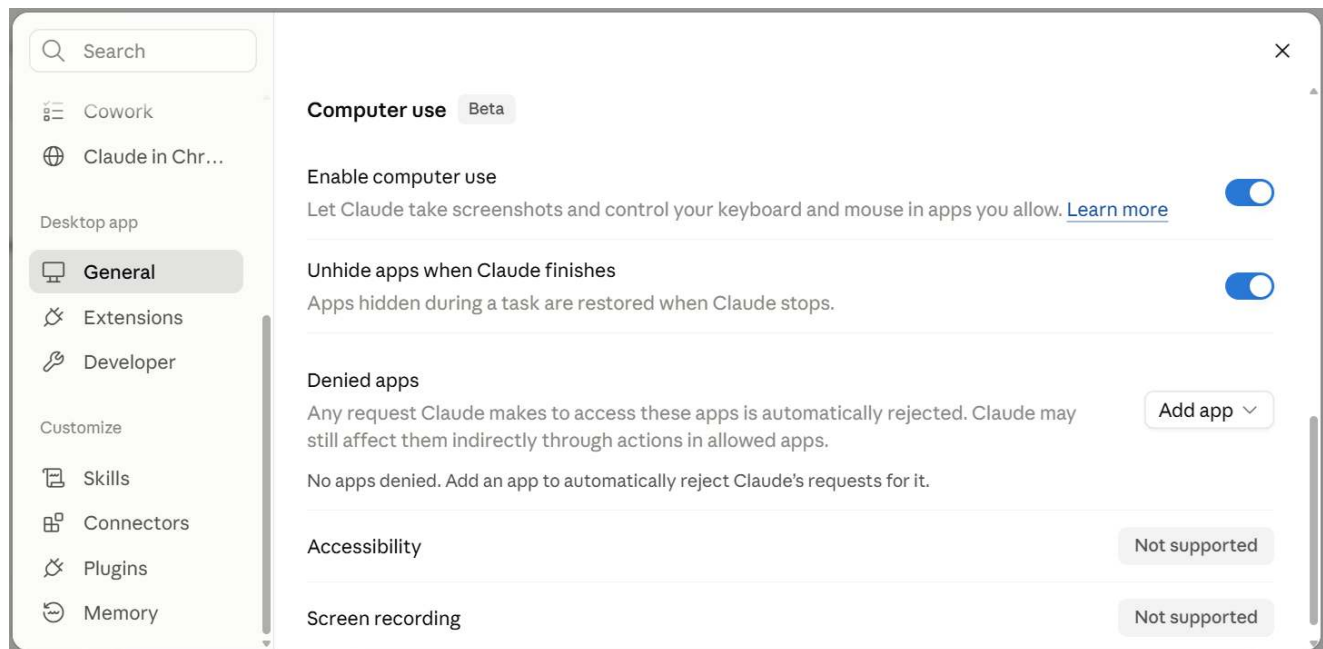


Figure 3.6B: Desktop app General settings and the Enable computer use toggle

Claude asks for permission before accessing each application. Some apps might be denied by default.

Automating Microsoft Paint on Windows or Preview on Mac

Anthropic describes computer use as giving Claude screenshot access plus mouse and keyboard control, and its Claude app release notes specifically say Claude can open files, point, click, and navigate what is on the computer.

To demonstrate computer use, we will control Microsoft Paint on Windows. Claude cannot accomplish the task merely by retrieving information. It has to operate a graphical desktop application: open Paint, interact with menus and dialogs, type values, choose a filename, and save the result.

Mac users can use Preview, which is included with macOS. Although Preview is primarily an image and PDF viewer, its Markup tools can add text, shapes, lines, arrows, signatures, crop and resize images, and draw freehand. The exact controls differ between the two apps, which makes this a useful computer-use demonstration: tell Claude the result you want rather than giving it a sequence of mouse clicks. Try following similar steps as for Paint on Windows but add a red rectangle around the heading and add the text “Reviewed” near the bottom-right corner.

Let’s try it:

1. Download the following sample image and save it in your Downloads folder: <https://github.com/markjprice/claude-vb/blob/main/Files/Book3/3-6-conference.png>. Its dimensions are 1672 x 941.
2. Open **Cowork** and ask Claude to perform a task that requires a desktop app like Microsoft Paint on Windows:

Open the image 3-6-conference.jpg from my Downloads folder in Microsoft Paint. Resize it so it is 1200 pixels wide while keeping the original proportions. Save the resized image as conference-photo-web.jpg in the same folder. Do not overwrite the original.



Watch out: Note that I have deliberately given the wrong file extension for the photo file that we want to edit. Let’s see how Claude handles this sort of common mistake.

3. When Claude asks for permission to access that app, review the request before allowing it only for this session:

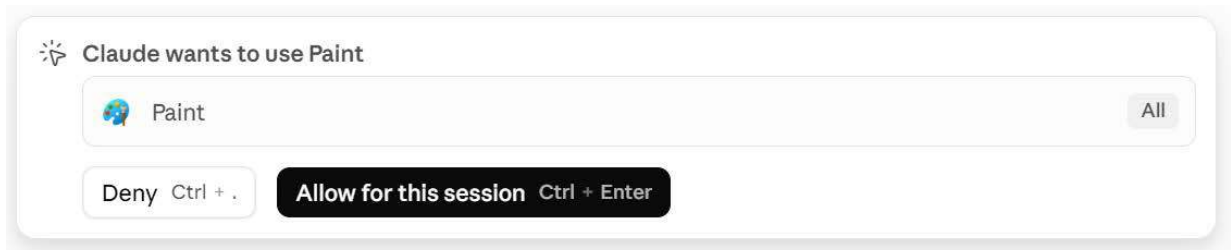


Figure 3.6C: Allowing computer use to control Microsoft Paint

4. Patiently watch your computer screen as Claude starts the app, opens the image file, adjusts the resolution, saves the file with a different filename, and checks that the file exists in the **Open** dialog box, taking screenshots along the way:

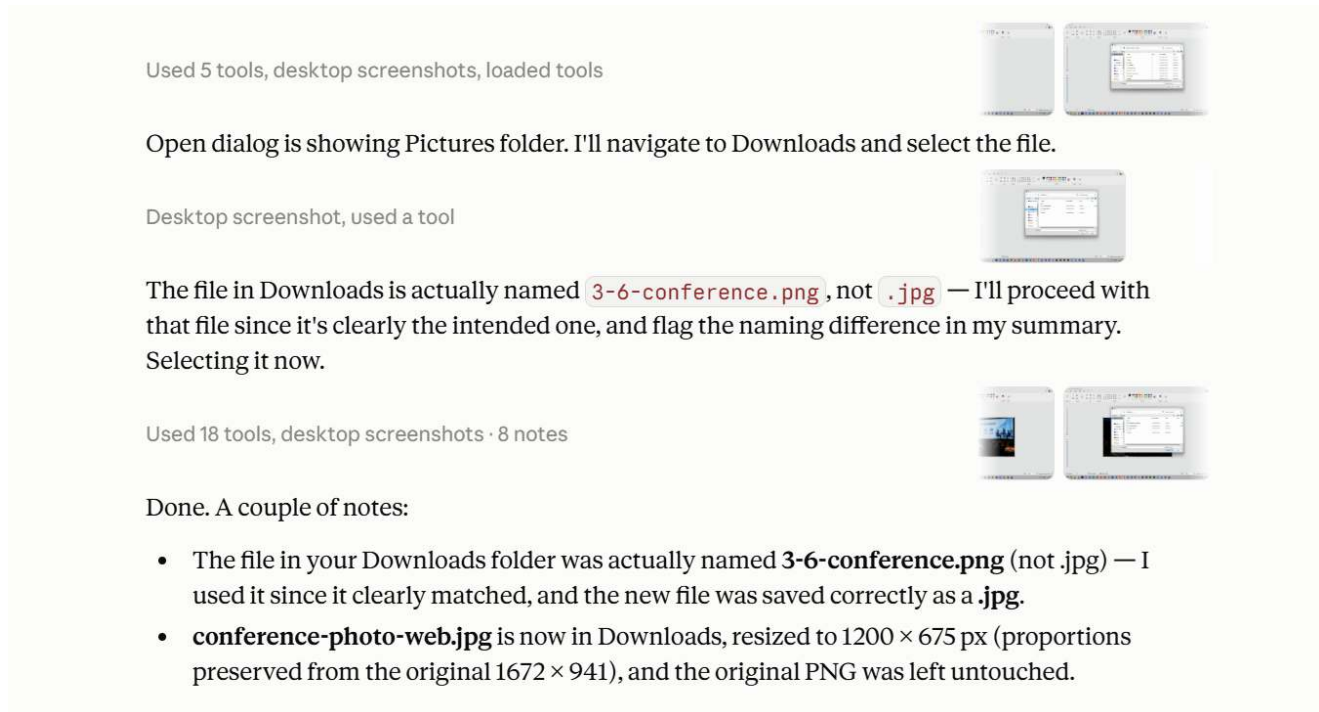


Figure 3.6D: Claude's summary and screenshots it took to complete the computer use task

Computer use can feel slow because Claude must take a screenshot after each action and review it to know what to do next. For example, you will see Claude hover the mouse pointer over screen elements and pause to take a screenshot to check that it is the correct button to click or that an option is selected correctly.



Watch out: Computer use requires the desktop computer to be awake and the Claude desktop app to be open. A Cowork task can continue in the cloud when your computer is off, but it cannot keep controlling apps on a computer it can no longer reach.

Automating Windows Explorer

Another example is controlling Windows File Explorer:

In my Downloads folder, create a folder called Conference Photos. Find the five JPEG files whose names start with IMG_Conference, move them into that folder, and rename them Conference-01.jpg through Conference-05.jpg. Show me what you plan to move and rename before making any changes.

For the preceding prompt example, Claude has to inspect the desktop environment, identify files, manipulate them, and respect an approval checkpoint.

Begin with a low-risk practice task

Your first task should be reversible and should not contain private or valuable information.

A simple practice is to use a temporary document and ask Claude to make a visible change in a desktop app. For example, create a disposable text document containing:

```
Practice Project
Owner: Alex
Status: Draft
Next review: September 15
```

Then ask Claude:

```
Use computer use to open the practice document in my default text editor.

Change:
Status: Draft

to:
Status: Ready for review

Do not change anything else.
Do not close the document or save it until I have checked the result.
```

This prompt does several useful things:

- It states which file and app behavior you want
- It defines exactly one change
- It says what Claude must not change
- It creates a human checkpoint before saving

Watch what Claude does. If it opens the wrong file or selects the wrong text, stop the task and correct it. The point of this exercise is not speed. It is to learn what appropriate supervision looks like.



Good practice: During your first few computer-use tasks, separate prepare from commit. Let Claude make or stage a change, then inspect it before allowing the final save, send, upload, or submission.

Give Claude boundaries before the first click

A vague instruction such as:

```
Clean up this customer database.
```

is dangerous because “clean up” could mean many things. Claude might delete duplicates, alter formats, merge records, or change information that only looks incorrect.

A better instruction defines the allowed area and the stopping conditions:

```
Open the Customer Records application and review the five records listed in
review-list.txt.
```

For each record:

- Check whether the postal code field is blank.
- If it is blank, add the postal code from approved-addresses.csv.
- Do not change any other field.
- If the customer is not present in approved-addresses.csv, make no change.

Process only those five records.

Before saving the first changed record, stop and show me what you intend to save.

After I approve the first one, continue with the remaining four.

Do not delete, merge, email, or export anything.

This is the same discipline you used when writing a skill. State the inputs, steps, rules, and acceptance criteria. Computer use adds one more requirement: define which actions need your approval:

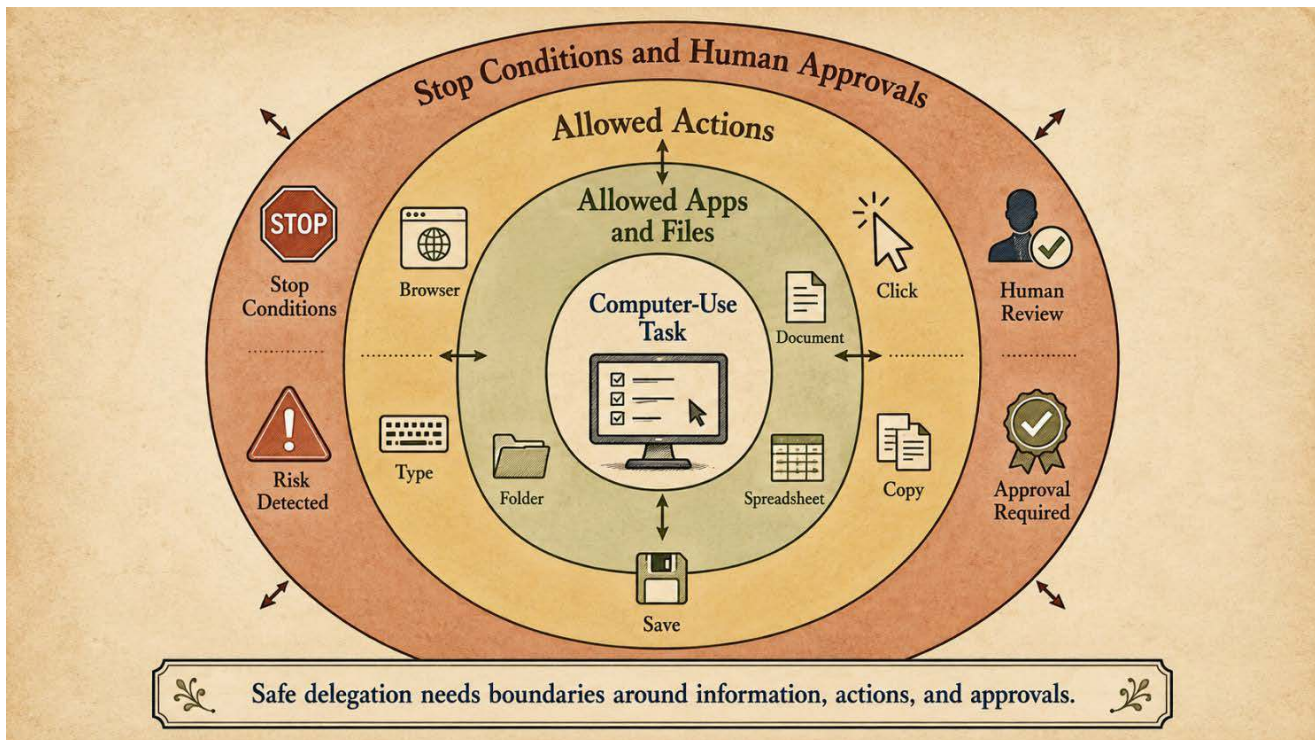


Figure 3.6E: Safe delegation requires boundaries around both information and actions

Understand what Claude can see

To control the screen, Claude takes screenshots and uses them to understand what is visible. If an approved app shows private information, Claude may be able to see that information too.

Before starting computer use:

- Close documents that are unrelated to the task
- Close private messages and personal accounts you do not need
- Avoid leaving passwords, API keys, financial details, or confidential records visible
- Use a test account or sample data when learning
- Give Claude access to the fewest apps needed for the job

The principle is simple: if Claude does not need to see something, do not leave it in the working area.

This matters especially when you ask Claude to browse the web and control desktop apps in the same task. Information from an untrusted website can contain hidden or misleading instructions designed to manipulate an AI system. This is called **prompt injection**.

A malicious webpage might contain text intended to persuade Claude to ignore your instructions, open another app, or reveal information. Anthropic builds defenses against these attacks, but those defenses are not perfect.



Watch out: Computer use has no sandbox between Claude's pointer and the applications on your screen. Limit what Claude can reach instead of relying on the model to refuse every dangerous instruction correctly.

Prefer read actions before write actions

A useful way to judge risk is to separate actions into two groups:

- Read actions gather information, like open a file, read a dashboard, inspect a form, compare values, or take notes
- Write actions change something, like type into a record, save a file, send a message, submit a form, delete or move information, or run a command that changes the system

Start a new computer-use workflow with read actions. Once you know Claude can navigate the app correctly, add a small write action that is easy to reverse.

For example:

```
Open the inventory application and inspect item A-1042.
```

```
Tell me:
```

- Current quantity
- Reorder threshold
- Supplier name

```
Do not edit any field.
```

Only after that works reliably should you consider a second task that updates a field:

```
Review the plan and steer while Claude works
```

Cowork shows what Claude is doing during a task. Use that visibility.

Before execution, check whether the plan matches your request. During execution, intervene if:

- Claude opens an unexpected app
- The visible record does not match the intended record
- A website asks for a new permission
- Claude is about to submit rather than preview
- The interface has changed and Claude appears confused
- The task reaches information that was outside your intended scope

Do not assume that a long plan must be completed once it begins. Steering is part of normal use.



Real-world example: An office manager asks Claude to copy ten approved maintenance dates from a spreadsheet into an old facilities-management program. Claude enters the first date correctly but then opens the wrong building record. The office manager stops the task immediately, points out the correct building, and asks Claude to restart from the second item. That is a successful use of supervision, not a failed automation.

Handle failures without repeating them blindly

Screen interfaces change. Buttons move. Pop-ups appear. An app may open a different window than expected. A computer-use task that worked yesterday can fail today.

When that happens:

1. Stop before the task makes more changes.
2. Identify the first step that went wrong.
3. Check whether the app or webpage changed.
4. Reduce the task to one smaller action.
5. Confirm the correct screen.
6. Try again only after you understand the failure.

Do not tell Claude simply to “try again” after a dangerous or unexplained error. That can repeat the same mistake faster.

If a task fails often, look for a better route. The software may now have a connector, an export feature, an API, or a more reliable file-based method.



Good practice: Treat computer use as a bridge to software without a better integration. If a trusted connector becomes available later, prefer it.

Know which tasks should stay manual

Computer use is not a good choice when a mistake could have immediate serious consequences.

Avoid using it unsupervised for tasks such as:

- Moving money or placing investments
- Purchasing expensive items
- Sending sensitive information
- Deleting large amounts of data
- Changing security settings you do not fully understand
- Making employment, legal, medical, or other high-stakes decisions
- Entering passwords or other secrets directly for Claude to see

Anthropic also trains Claude to avoid some risky actions, including stock trading and entering sensitive data. Those safeguards help, but they are not a substitute for your own access controls. The safest automation is often one that prepares the work but leaves the final action to you.

For example, Claude can open a web form, fill in non-sensitive fields, and stop before submission. You inspect the form and select Submit yourself.

Combine a skill with computer use

A skill and computer use solve different problems. A skill tells Claude how the procedure should be done. Computer use gives Claude a way to interact with software that lacks a better interface.

Suppose an HR team has an old training-record application. You could create a Training Record Update skill containing rules such as:

- Update only completed courses
- Use the completion date from the approved certificate
- Never change the employee's name or identifier
- Stop if the certificate and employee record do not match
- Produce a change log

Computer use can then carry out the visible steps in the application. The skill supplies the rules; computer use supplies the hands. This combination is more reliable than giving Claude a vague instruction each time.



Learn more online: *Let Claude use your computer in Cowork:* <https://support.claude.com/en/articles/14128542-let-claude-use-your-computer-in-cowork>.

Try it now

Choose one desktop program you use that does not have a connector and that contains no sensitive information you need for this exercise. Create or use a disposable test file or test record. Ask Claude to perform one small visible change, but require it to stop before the final save or submission. Then repeat the task with a second test record.

Do not move on to a larger task until both tests work as expected.



Watch out: Avoid tasks involving online banking, purchasing, sending email, deleting files, or changing security settings at this point. They will distract you from the concept by making you worry about risk instead of understanding what computer use actually does.

Check the result

- ☐ Did Claude use a connector or browser tool instead when a more precise option was available?
- ☐ Did you approve only the app needed for the task?
- ☐ Could Claude see unrelated private information on your screen?
- ☐ Did your prompt state what Claude was allowed to change and what it must not change?

- ☐ Did you include a checkpoint before a consequential action?
- ☐ Did Claude navigate to the correct record or document twice in a row?
- ☐ If something failed, did you identify the cause before trying again?
- ☐ Would a connector, file tool, or ordinary manual process be more reliable for this task?

Computer use lets Claude reach software that connectors cannot. The next chapter combines the building blocks you have learned so far into useful personal and work workflows.

Chapter 3.7 Automate Personal Organization and Build Workflows

Book 3 has covered the building blocks of automation. This chapter combines them into practical personal and workplace workflows. You do not need every building block in every workflow. Start with the smallest combination that solves the problem reliably.

Build a recurring personal-organization workflow

Many personal tasks repeat on a schedule: a weekly review of your calendar, a monthly budget check, or a seasonal home-maintenance list. These are strong candidates for the scheduled tasks you met in *Chapter 3.2*, combined with a project that holds your ongoing context:

1. Choose one recurring personal task, such as planning your week.
2. Create a project and add the information Claude needs each time, such as your priorities or recurring commitments.
3. Write a short skill or standard prompt that produces the result you want, such as a prioritized task list.
4. Schedule it to run at a time that fits your routine.

Remember the difference between projects and skills:

- **Project:** keeps the subject, files, ongoing context, and project-scoped memory
- **Skill:** stores the reusable lesson procedure

Turn a learning goal into a repeatable routine

Chapter 1.6 showed you how to learn a subject with Claude in a single conversation. You can turn that same approach into a routine you repeat over weeks, rather than a one-time session:

- Keep one project for the subject, with your current level and goals stated once
- Use a simple skill that asks Claude to explain one new idea, give an example, and quiz you on the last session's material
- Review your progress every week or two, and adjust the pace based on what is actually sticking



Real-world example: A retired teacher learning to use a spreadsheet program keeps a single project called *Spreadsheet basics*. Each session, she asks for one new skill, a short practice task, and two questions that check whether she remembers the previous lesson. After six weeks, she can build a simple household budget without help.

Prepare for a personal decision or milestone

Some personal automation is not about repeating a task, but about supporting a single important moment, such as a big purchase, a move, or a conversation you are nervous about. *Chapter 1.7* covered comparing options for everyday decisions. You can extend that into a short, repeatable pattern:

1. State your situation, goals, and constraints once in a project.
2. Ask Claude to list the questions you still need answered.
3. Gather information over several sessions, adding it to the same project.
4. When ready, ask for a clear summary of your options and open questions, not a final decision made for you.

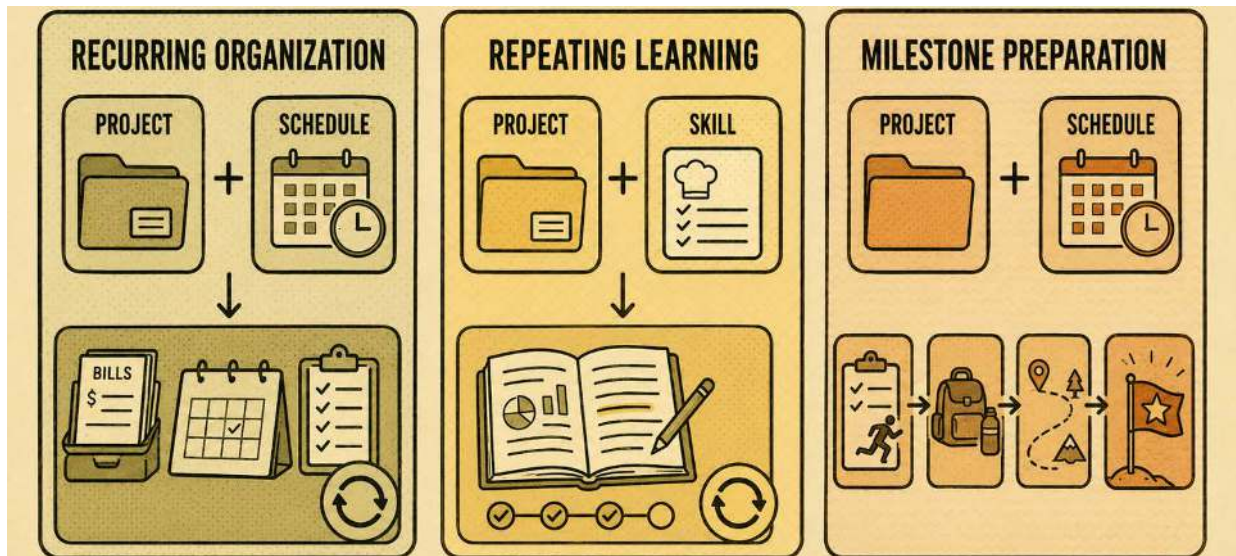


Figure 3.7A: The same combination of a project and a small skill or schedule can support ongoing organization, steady learning, or preparation for one big decision



Watch out: Keep personal and sensitive information, such as financial details or health notes, in a project you control, and review your privacy and memory settings from *Chapter 1.11* before adding anything you would not want stored longer than necessary.

Build a Research and Content Workflow

This chapter builds a complete workflow around one common need: turning research into finished written content on a repeating basis. It combines the skills, scheduling, and connections from earlier chapters in *Book 3* into one working pattern you can adapt to your own content.

Produce a recurring management review

A management review is a short, regular report that turns raw information into a decision-ready summary. Building it as a workflow, rather than a fresh request each time, keeps the format consistent and saves you the setup work every week:

1. Define what the review must cover, such as key metrics, notable changes, and open risks.
2. Connect the sources it needs, using the plugins from *Chapter 2.8* or the MCP connections from the *Chapter 3.5*.

3. Build a skill that gathers the information and drafts the review in your required format.
4. Schedule the workflow to run before your regular reporting deadline, as you learned in *Chapter 3.2*.

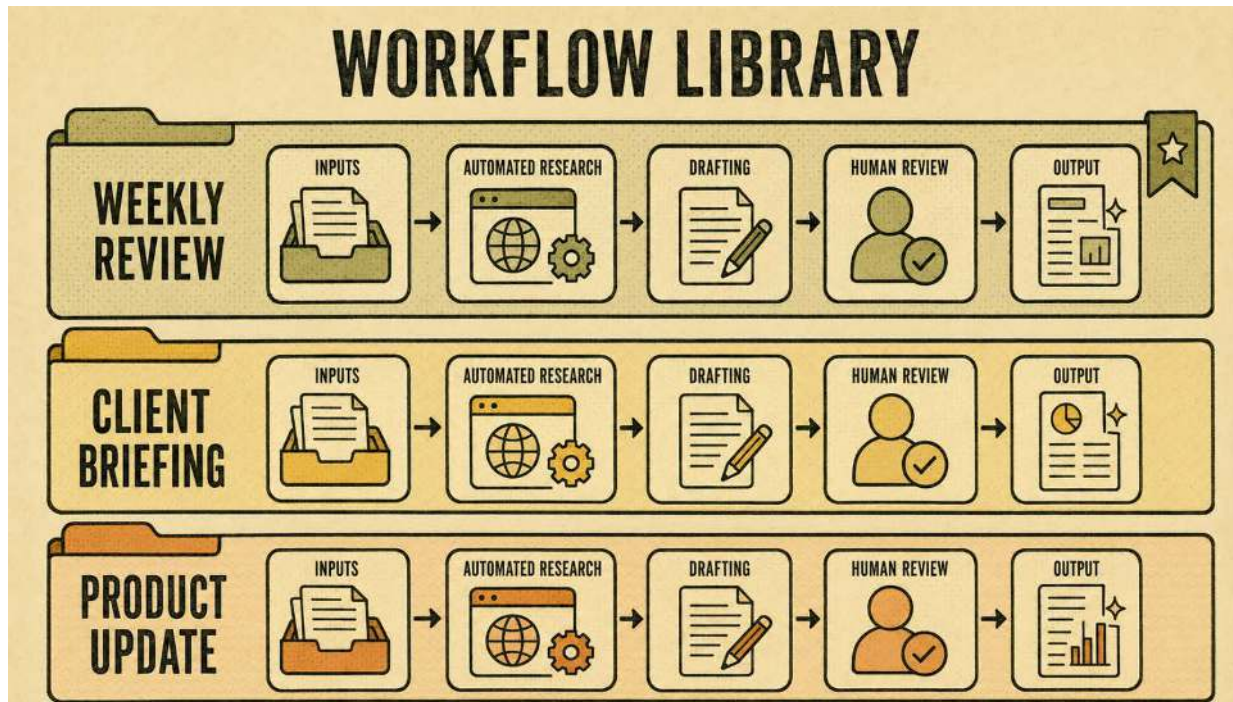


Figure 3.7B: A workflow library holds several small, repeatable patterns, each moving from inputs through automated stages to a reviewed output

Gather and compare evidence before drafting

Strong content depends on strong evidence gathered first. Before drafting, have Claude collect relevant facts, figures, and sources, then compare them for agreement or conflict, the way you learned in *Chapter 1.9*'s deeper research process:

- Ask for at least two independent sources for any important claim
- Flag any number or claim that sources disagree about, rather than picking one silently
- Keep a short list of sources alongside the draft, so a reader or colleague can check them later



Watch out: A workflow that drafts content automatically can also repeat an error automatically, every time it runs. Check the evidence stage carefully the first several times you run a new workflow, before you start trusting it without review.

Develop options and recommendations

Some content workflows should end in a recommendation, not only a summary. When that is the goal, ask Claude to present two or three real options with their trade-offs, rather than jumping straight to one preferred answer.

1. State the decision the content should support:
2. Ask for the leading options, each with its evidence, benefits, and drawbacks.
3. Ask for a recommendation, along with the reasoning behind it.
4. Review the recommendation against the evidence before you rely on it.



Good practice: Keep one part of every content workflow manual: a final human read-through before the content goes to its audience. Automation should remove repetitive drafting work, not remove the last check.

Build a Decision and Operations Workflow

This chapter builds a workflow that watches ongoing situations, flags what needs attention, and prepares you for a specific decision or action. The pattern is the same set of building blocks from earlier in *Book 3*, applied to operations instead of content.

Monitor decisions, risks, and follow-up actions

Many roles involve tracking things that change: open decisions waiting on someone, risks that need a response, and actions that were promised but not yet done. A monitoring workflow keeps this list current without you rebuilding it from memory each week:

1. Define what counts as an open decision, a risk, or a follow-up action for your work.
2. Connect the sources where these items appear, such as notes, messages, or a shared tracker, using an approved connector. If the system is not in Claude's connector directory, *Chapter 3.5* showed how a trusted remote MCP server can provide a custom connector.
3. Build a skill that reviews these sources and produces a short, current list.
4. Schedule the workflow to run before the meeting or checkpoint where you need the list.

Prepare for prospect and account conversations

Sales and business-development work often depends on a short brief prepared before a conversation. This is a strong candidate for the same pattern: gather relevant information about a prospect or account, then produce a consistent brief every time:

- Gather public information about the prospect or account, along with any internal notes you have permission to use
- Summarize what matters for the upcoming conversation: their situation, likely needs, and open questions
- Flag any information that could not be confirmed, rather than presenting a guess as a fact



Watch out: Never include personal or confidential information about a prospect that you do not have a legitimate reason to hold or share. Review *Chapter 2.1*'s rules on data classification before connecting any customer information source.

Prepare for screening and interview conversations

The same brief-before-conversation pattern applies to hiring. Before a screening call or interview, a short workflow can compare a role's requirements against a candidate's stated experience and prepare focused questions:

1. State the role's key requirements clearly.
2. Provide the candidate's resume or application material.
3. Ask Claude to identify where the match looks strong, where it looks uncertain, and what to ask about directly.

4. Review the questions before the conversation, and remove anything that could introduce unfair bias.

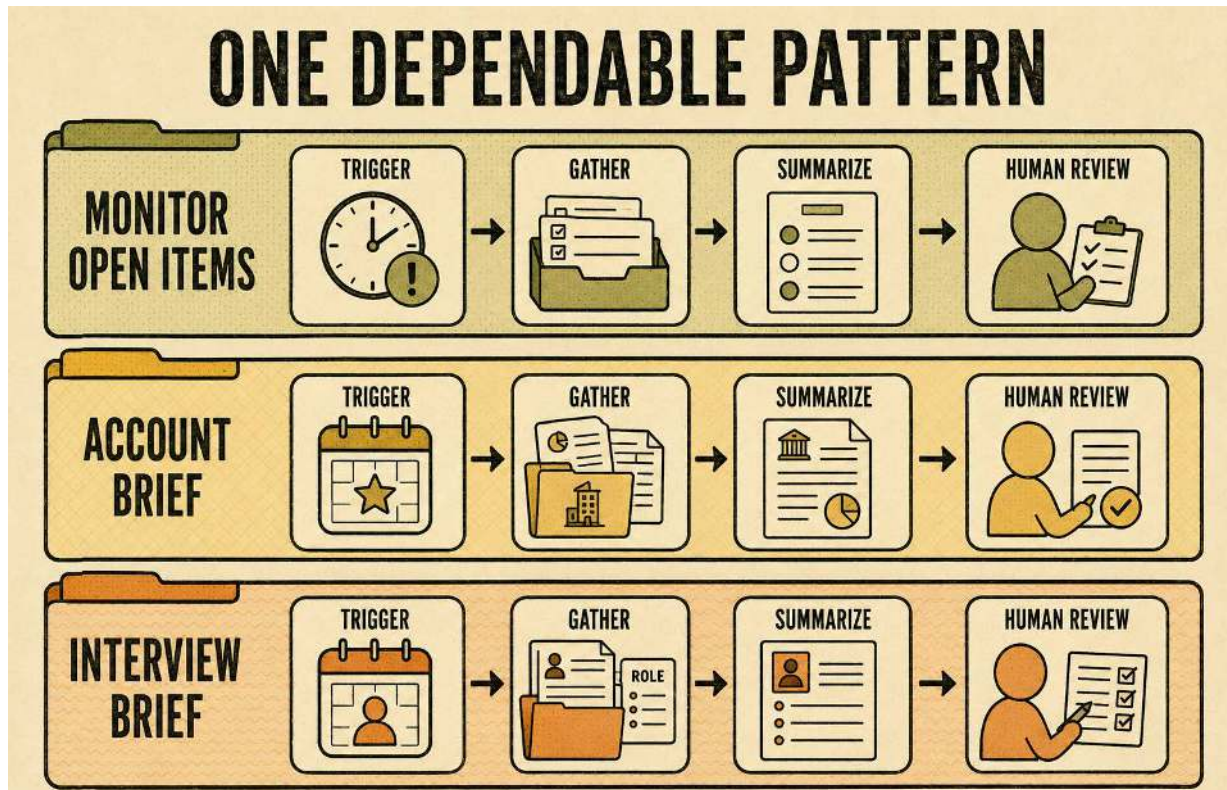


Figure 3.7C: Monitoring, account preparation, and interview preparation all follow the same pattern: a trigger, a gathering step, a summary, and human review before the conversation



Good practice: Keep a fixed set of questions or criteria for every screening brief, rather than letting the workflow generate new ones each time. Consistency protects fairness as much as it protects your time. Claude should summarize job-related evidence and prepare consistent questions rather than automatically rank or reject candidates.

Try it now

Pick one recurring personal task or one learning goal. Set up a small project for it, with one skill or standard prompt you can reuse. Run it once this week.

Pick one recurring report or piece of content you produce regularly. Sketch the sources it needs, the evidence-gathering step, and whether it should end in a summary or a recommendation. Build the first version as a skill or a small scheduled workflow.

Choose one operational task you repeat before a meeting or a conversation, such as a status check, an account brief, or a screening summary. Build the first version as a skill that gathers the relevant information and produces a consistent result.

Check the result

- ☐ Did the routine save you real time or effort compared with starting from scratch each time?
- ☐ Is the information stored in the project something you are comfortable keeping there?

- ☐ Would you keep using this routine next month, or does it need a change first?
- ☐ Does your workflow gather evidence from more than one source before drafting?
- ☐ Did you build in a step to flag disagreement between sources, rather than hiding it?
- ☐ Is there still a human read-through before the content reaches its audience?
- ☐ Does the workflow's list or brief match what you would have prepared by hand?
- ☐ Did you check for any confidential or unfair content before using the result in a real conversation?
- ☐ Would this workflow still work if the source information changed shape next month?

With a personal routine automated, you applied the same thinking to work you do for other people by building a workflow around producing content, from research through a finished draft, and a workflow for decisions, and daily operations.

Chapter 3.8 asks you to choose one of these workflows, or a new one, and prove it holds up: measured against a baseline, tested with a case designed to fail, and ready to run without you standing over it.

Chapter 3.8 Project – Build and Measure an End-to-End Workflow

This project brings together the main building blocks from *Book 3*: choosing the right mechanism, building and testing a skill, scheduling repeated work when useful, adding a trusted connector when outside information is required, and keeping human approval before consequential actions.

Choose a repeated workplace process

Pick one process from your own work that repeats often enough to matter, and that you understand well enough to describe clearly. A good candidate has a defined start, a defined end, and a result you can judge as good or not good enough.

Avoid choosing your hardest or most sensitive process for a first attempt. Choose one where a mistake is easy to catch and easy to fix.

Combine projects, skills, scheduled tasks, and connectors

Use the tools from this book together, choosing only what the process actually needs:

1. Set up a project if the workflow needs ongoing context, as in *Chapter 1.11*.
2. Add a scheduled task only if the workflow should run repeatedly, as in *Chapter 3.2*.
3. Build one or more skills for the stable, repeated procedures, as in *Chapter 3.3* and *Chapter 3.4*.
4. Add a connector only if the workflow needs outside information or action, as in *Chapter 3.5*.
5. Use computer use only if the workflow must operate software that does not have a suitable connector, browser integration, or direct file method, as in *Chapter 3.6*. If you use it, include an approval checkpoint before the first write action.



Figure 3.8A: A complete workflow moves from trigger through intake, specialist work, quality review, and approval, ending in delivery and a record of what happened

Add review and approval checkpoints

Every workflow in this project needs at least one point where a person checks the result before it goes anywhere consequential. Decide where that checkpoint belongs, and what the reviewer should specifically look for.

State clearly when the workflow should stop and ask a person, rather than continuing on its own. A workflow that cannot recognize its own limits is not ready to run without close supervision.

Test, document, and reuse the workflow

Follow this sequence to turn your workflow from a working prototype into something you can trust and reuse.

1. Record the existing time and quality baseline, before your workflow existed.
2. Define what success looks like in specific, checkable terms.
3. Test the workflow on a normal case, an unusual case, and a case designed to fail.
4. Confirm exactly when the workflow stops and asks a person for input.
5. Record what happened during each test, including any surprises.
6. Compare the result against your baseline from step one.
7. Provide a manual fallback in case the workflow is unavailable or fails.



Good practice: Keep this record. *Chapter 4.3* shows you how to use evidence like this to document your improvement and build a case for the value of your work.

Choose your version of the project

Pick the version that fits your situation best.

- A personal or family workflow, such as an organizing or planning routine
- A content or research workflow, such as the recurring review
- A decision or operations workflow, such as the monitoring or briefing routine

Try it now

Complete your chosen workflow this week. Run it through every step above, from choosing the process to recording your test results and comparing them against your baseline.

Check the result

- ☐ Does your workflow perform measurably better, the same, or worse than your baseline?
- ☐ Did every test, including the one designed to fail, behave the way you expected?
- ☐ Do you have a manual fallback ready if the workflow cannot run?

The next step

A workflow that runs on its own, passes its tests, and beats your baseline is a genuine capability, not a demo. *Book 4, Advance Your Career and Income*, shows you what to do with that capability. You'll learn to measure the value of what you've automated, build a case for a promotion or a new role, and decide whether the skills and workflows you've built are worth turning into a service or product of your own.

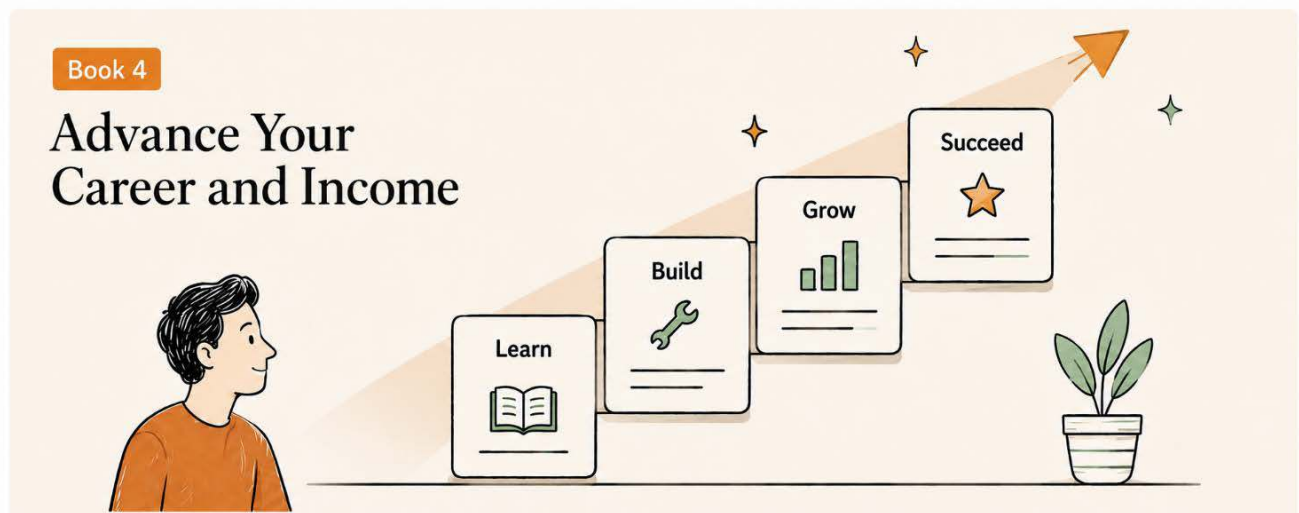
Book 4

Advance Your Career and Income

In *Book 4*, you will use what you learned in *Book 1*, *Book 2*, and *Book 3*, to earn a promotion, make a career change, or provide a service to generate additional income.

It has eleven chapters, including an end-of-book project:

- 4.1 Choose What Better Work Means to You 182
- 4.2 Audit Your Role and Find High-Value Work 184
- 4.3 Measure and Document Your Improvement 186
- 4.4 Learn the Skills Your Next Role Requires 189
- 4.5 Combine AI Fluency with Human Strengths 191
- 4.6 Propose and Lead an AI Improvement at Work 193
- 4.7 Build a Case for Promotion or Higher Pay 195
- 4.8 Prepare for a Better Job or Career Change 198
- 4.9 Build a Portfolio and Measurable Case Study 200
- 4.10 Turn a Problem into a Service or Product 203
- 4.11 Project – Complete a 30-day Career Value Project 205



Chapter 4.1 Choose What Better Work Means to You

Book 3 helped you build workflows and measure whether they worked. *Book 4* asks a different question: what should you do with the ability you have built? This chapter helps you choose a direction before you invest more time.

There is no single right answer. Two readers with the same skills might choose completely different paths, and both choices can be good ones.

Your current role — Reduce routine work

Some readers mainly want their week to feel less repetitive. If routine tasks take up hours you would rather spend elsewhere, the goal is to hand over more of that routine work to Claude, using the automation skills and processes from *Book 3*.

This is often the fastest path to a better week, since you already know which tasks feel repetitive and draining.

A promotion — Increase impact or earn higher pay

If your goal is a better title or a higher salary in your current organization, later chapters in *Book 4* show you how to document your improvement and build a case that colleagues and managers will find convincing.

A new role or career change — Develop expertise

Other readers have enough time but want their work to matter more. Here, the goal is not doing less, but redirecting freed-up time toward work that produces a bigger result, such as a project that reaches more people or solves a harder problem.

To go deeper in your field rather than wider, you can develop your existing skills. Claude can help you learn faster, as you saw in *Chapter 1.6*, so the time you save on routine tasks can fund real study in your specialty.

Some readers want to move into a different role or a different field entirely. *Chapter 4.4* and *Chapter 4.8* focus directly on this path, from learning new skills to preparing your job search.

Earn independently — Sell a product or service

A smaller group of readers want to turn their abilities into a service or product they sell directly, whether alongside their current job or instead of it. *Chapter 4.10* and *Chapter 4.11* cover this path in detail, without promising quick or guaranteed income.



Watch out: Earning independently takes real work: finding customers, delivering consistently, and handling problems when they come up. Treat any promise of quick or easy income with suspicion, including promises made by Claude itself.

Decide where saved time should go

Whatever path you choose, the time and mental energy you free up through automation will not automatically go somewhere useful. You have to decide.

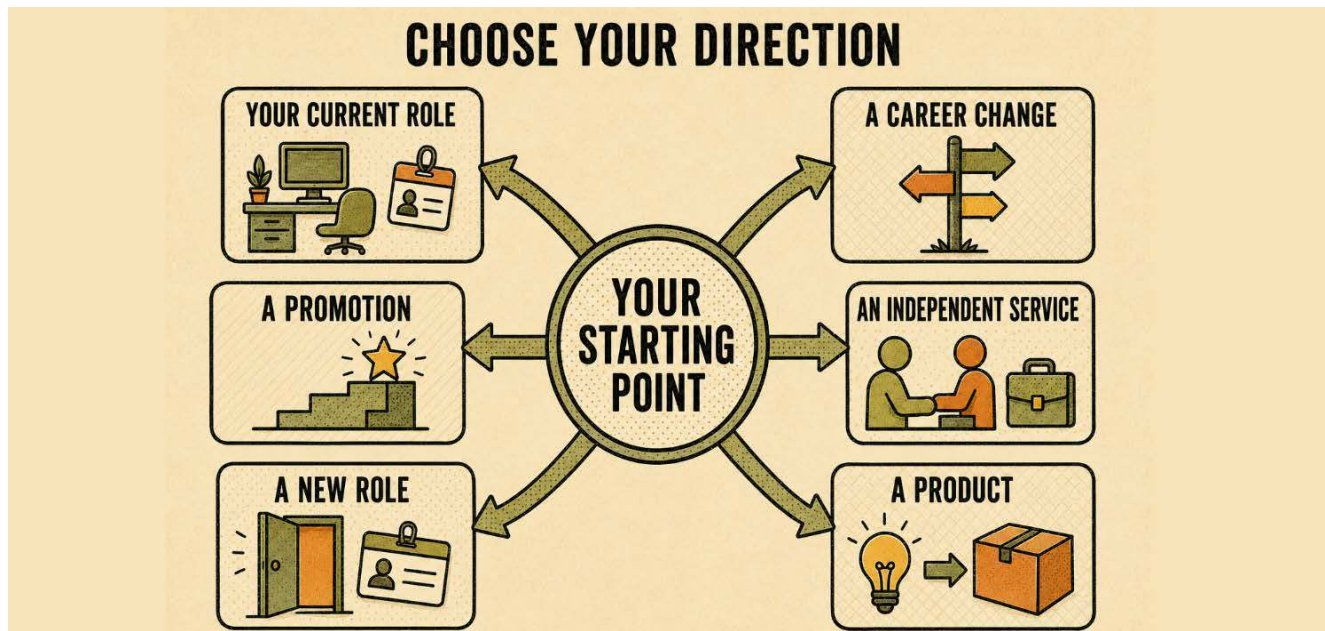


Figure 4.1A: Better work can mean many different things. This map shows common directions readers take once they have freed up time and built new skills.

Before moving on, write down which direction, or combination of directions, matters most to you right now. You can change your mind later. *Book 4* works best when you have a working answer to start from.



Good practice: Choose one primary direction, even if you are curious about several. Chapters that follow build on a specific choice, and a scattered effort across many directions tends to produce less progress than a focused one.

Try it now

Read the mapped directions in this chapter again. Rank your top two. Write one sentence about why each one matters to you, using your own words rather than the wording in this book.

Check the result

- ☐ Did you choose a direction based on what you actually want, rather than what sounds most impressive?
- ☐ Can you explain your choice to another person in one or two sentences?
- ☐ Are you comfortable spending the next several chapters working toward this direction?

This chapter helped you choose a direction. *Chapter 4.2* turns that direction into a specific starting point: one piece of your current work worth improving first.

Chapter 4.2 Audit Your Role and Find High-Value Work

This chapter focusses on one piece of your current work worth improving.

List recurring activities

Start with a plain list of what you actually do. Over two or three days, write down each task as you do it, including small ones you might normally overlook, such as status updates or routine replies.

You do not need a complete list on the first try. A rough list you can refine is more useful than a perfect list you never finish.

Identify friction and wasted effort

Look at your list and mark the tasks that feel slow, repetitive, or frustrating compared with the value they produce. Common signs of friction include retyping the same kind of message, searching for the same information more than once, and waiting on a step that could run without you.



Real-world example: A project coordinator notices she rewrites the same weekly status update from scratch each Friday, pulling numbers from three different places. The task itself takes thirty minutes, but finding the numbers takes closer to an hour. That gap between the task and the effort around it is a clear sign of friction.

Classify work as retain, assist, automate, delegate, or stop

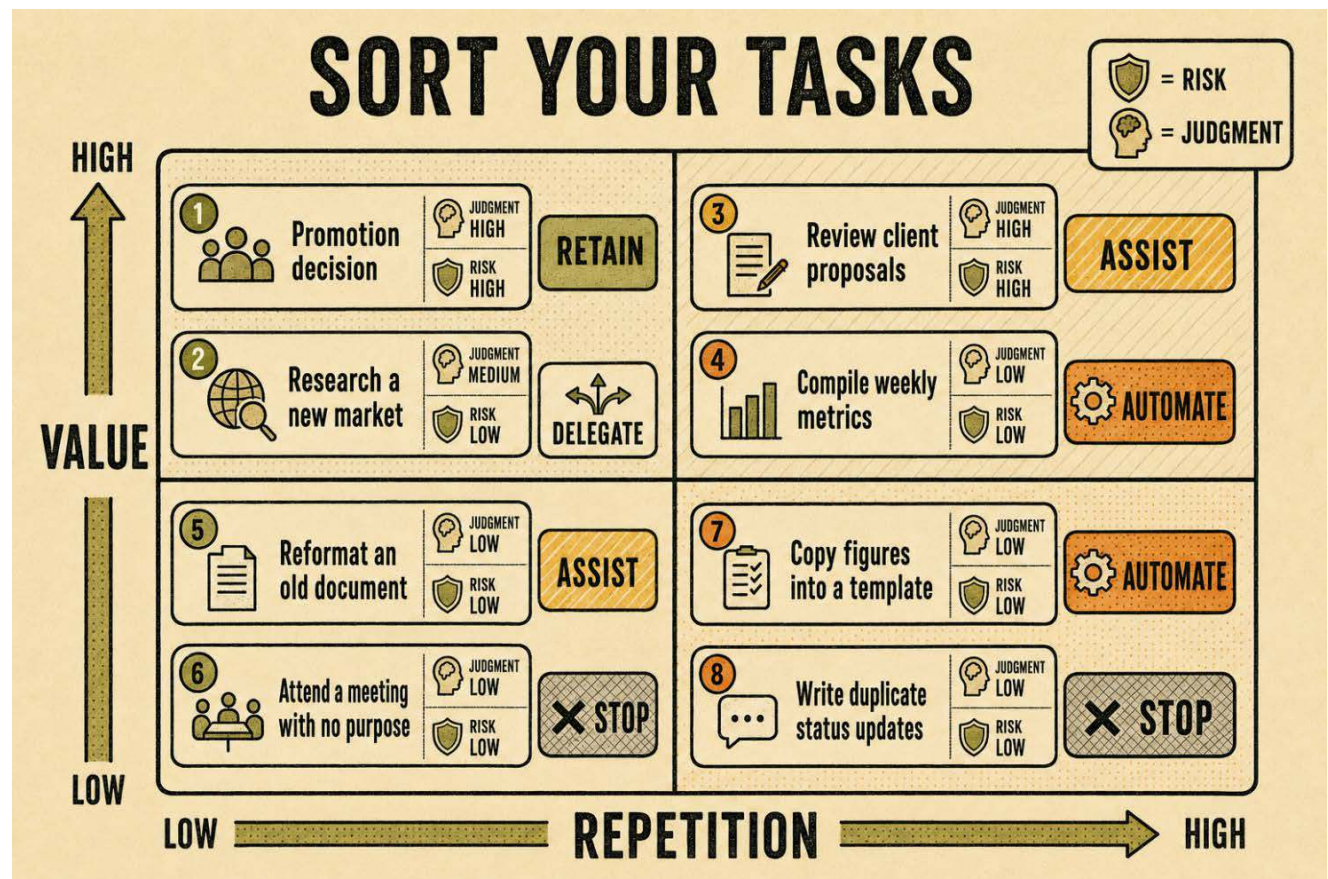
Sort each recurring activity into one of five categories:

1. **Retain:** work that depends on your judgment, relationships, or authority, and should stay with you as it is
2. **Assist:** work you will keep doing, but where Claude can help you go faster or check your work
3. **Automate:** work stable and well understood enough to become a skill or workflow, as in *Book 3*
4. **Delegate:** work better handled by a colleague, a specialist subagent, or another person entirely
5. **Stop:** work that no longer serves a real purpose and could end

Some tasks should not move to automate or delegate even if they repeat often, because a mistake would be costly or hard to reverse. This includes work involving confidential information, financial commitments, legal obligations, or decisions that affect another person's job or safety.



Watch out: A task that looks routine on the surface can still carry real risk. Before automating anything, ask what happens if the result is wrong and nobody catches it. If the answer is serious, keep a human check in place, even after you build a workflow for the rest of the task.



Choose one improvement

1. Review your classified list.
2. Cross out anything marked retain or stop, since those need a decision rather than a build.
3. From the remaining tasks, choose the one with the most friction and the least risk.
4. Write one sentence describing what “better” would look like for this task.



Good practice: Write your classification down somewhere you can revisit, not only in your head. You will use this list again in later chapters as you look for the next task to improve.

Try it now

Spend fifteen minutes listing your recurring activities. Classify at least five of them using the five categories in this chapter, then choose the one task you will improve first.

Check the result

- ☐ Does your chosen task show clear friction, real repetition, and manageable risk?
- ☐ Did you classify at least one task as retain, so you can see the categories are not only a path toward removing yourself from your own work?
- ☐ Can you describe what a better version of this task would look like, in one sentence?

Finding a high-value task to improve is only useful if you can prove the improvement happened. *Chapter 4.3* shows you how to measure your starting point and document real progress over time.

Chapter 4.3 Measure and Document Your Improvement

In *Book 3*, the end-of-book project asked you to record a baseline before building your workflow. This chapter shows you how to use that kind of evidence well, whether it comes from a *Book 3* process or a fresh improvement you are testing now.

Evidence matters because a claim of improvement without evidence is just an opinion, and opinions are easy to doubt. A specific measurement is much harder to argue with.

Establish a baseline

A baseline is a clear record of how a task worked before your improvement. Without one, you cannot honestly say whether anything actually got better:

1. Choose the task you identified in *Chapter 4.2*.
2. Note how long it currently takes, from start to finish.
3. Note its current quality, using a specific measure such as an error rate or a completion rate.
4. Record this baseline before you change anything.

Measure time, quality, errors, and outcomes

Once you have a baseline, measure the same task after your improvement, using the same measures each time so the comparison is fair:

- **Time:** how long the task takes from start to finish
- **Quality:** how often the result meets your stated standard
- **Errors:** how often something needs correcting after the fact
- **Outcomes:** the real-world result the task was meant to produce, such as a client's satisfaction or a report's usefulness to its reader

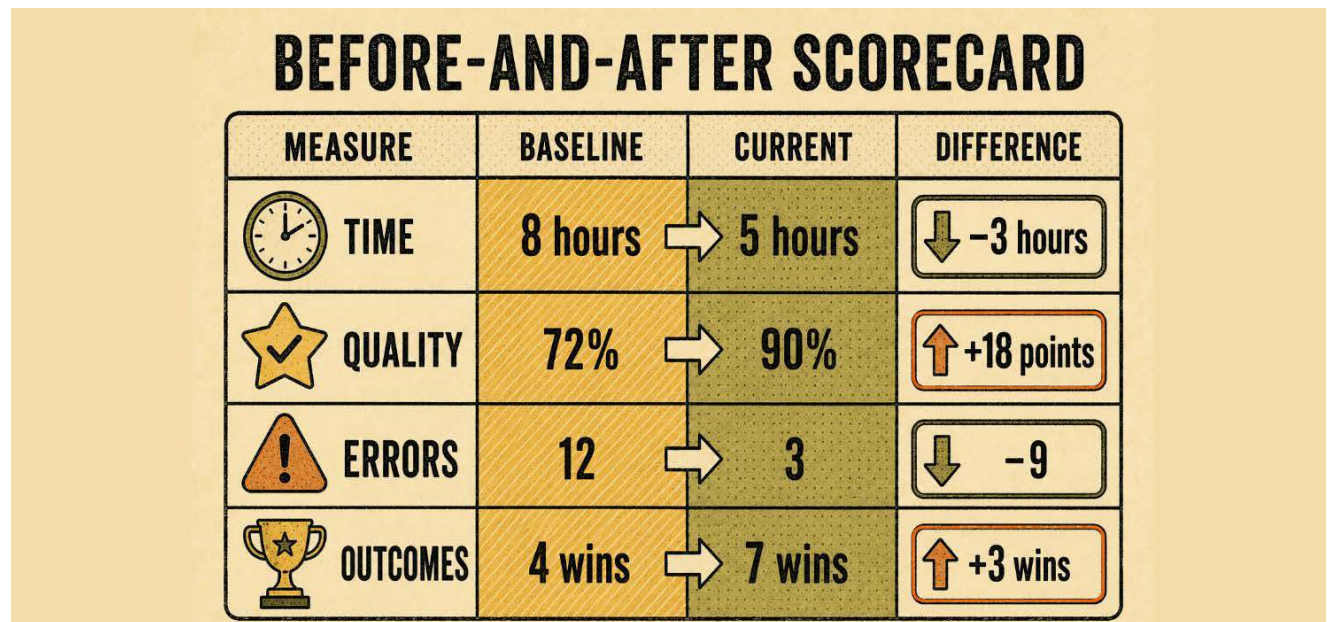


Figure 4.3A: A before-and-after scorecard puts your baseline and your current result side by side across time, quality, errors, and outcomes

Avoid invented precision

Report your measurements honestly, even when they are rough. If you only have a general sense that a task now takes about half the time, say that, rather than presenting a precise number you did not actually measure.



Watch out: A specific-sounding number built on a guess is more misleading than a stated estimate. If someone asks how you got a number, you should be able to explain it. If you cannot, round it or describe it as an estimate instead.

Keep an achievement record

Do not rely on memory to carry your evidence forward. Keep a running record of every task improvement you make, with its baseline, its result, and the date:

1. Create one document or spreadsheet for your achievement record.
2. Add a new entry each time you complete an improvement, including the ones that did not work as well as you hoped.
3. Include enough detail that you could explain the entry to someone else a year from now.

- Review the record occasionally and update it as circumstances change.

Achievement Record

A running log of measured improvements: what you improved, where you started, what you achieved, and when.

Task improvement	Baseline	Result	Date
Weekly status report	90 min total: 30 min writing, ~60 min gathering numbers from three sources	35 min total, using a saved Claude prompt to draft from pasted data	14 Mar 2026
Client onboarding emails	~45 min per new client, rewritten from scratch each time	15 min per client, using a reusable template Claude helped refine	02 Apr 2026

Figure 4.3B: An achievement record collects your improvements over time, so you always have evidence ready when you need it



Sample files: A one-page Word achievement record: <https://github.com/markjprice/claude-vb/tree/main/Files/Book4/4-3-Achievement-Record.docx>. The table has four rows drawn from the book's running example, including one entry that only partly succeeded.



Good practice: Include improvements that did not fully succeed in your record, along with what you learned. A record of only perfect results looks less credible than one that also shows honest attempts.

Decide how to use the capacity created

Once you have measured a real improvement, decide what to do with the time or effort it freed up. Return to *Chapter 4.1* and check that the capacity is going toward the direction you chose there, rather than disappearing into other routine work.

Try it now

Measure your chosen task's baseline and its current result, using the four measures in this chapter. Add the comparison to a new achievement record, even if the record only has one entry so far.

Check the result

- ☐ Can you explain exactly how you measured each number in your comparison?
- ☐ Did you record the result honestly, including any part of the improvement that fell short?
- ☐ Do you know where the time or effort you saved is now going?

Documenting your current improvement builds a record for today's role. *Chapter 4.4* looks further ahead, helping you identify and learn the skills your next role will actually require.

Chapter 4.4 Learn the Skills Your Next Role Requires

If *Chapter 4.1* pointed you toward developing expertise or changing career, this chapter helps you find out exactly what to learn, then build a plan you will actually follow.

Research how a role is changing

Roles change as tools, expectations, and industries change. Before setting a learning plan, research what your target role looks like today, not what it looked like when you first learned about it:

1. Ask Claude to search for current descriptions of your target role, including recent job postings.
2. Ask it to identify skills that appear often across several sources.
3. Note any skill that seems new or newly emphasized compared with older descriptions of the same role.
4. Check a few of the sources yourself, using the research habits from *Chapter 1.8* and *Chapter 1.9*.

Compare current and required skills

List the skills your target role needs, then honestly rate your current level in each one. Be specific rather than general. “Comfortable with data” is harder to act on than “can build a pivot table but not a formula-driven dashboard.”

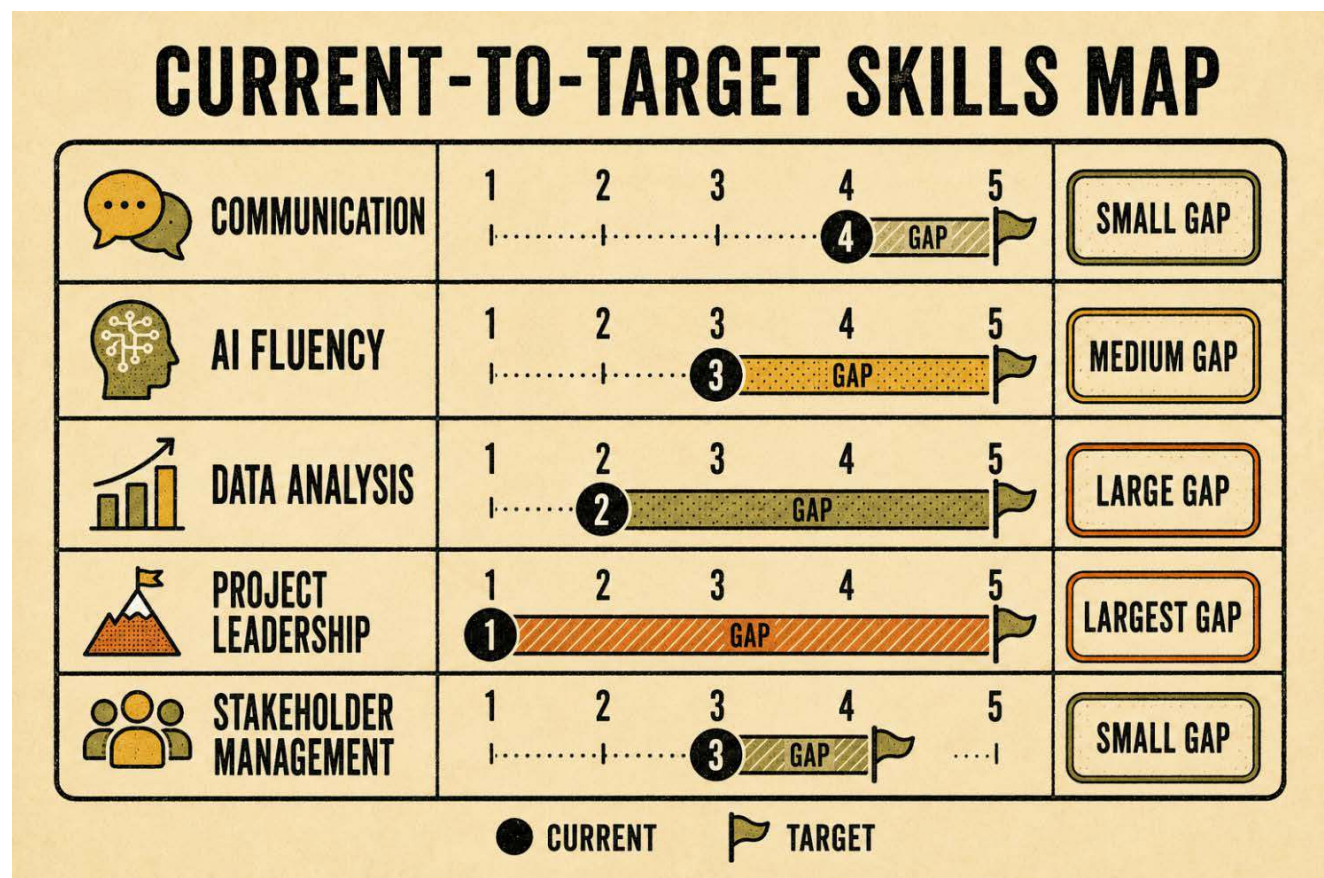


Figure 4.4A: A current-to-target skills map shows exactly how far you are from the skills your next role needs, one skill at a time

Create a learning plan

Turn your biggest gaps into a plan with real steps and a real timeline. *Chapter 1.6* showed you how to learn a single subject with Claude. Extend that approach across several linked topics, in an order that builds on itself:

1. Choose two or three skills to focus on first, based on the size of the gap and how much the role depends on them.
2. Ask Claude to break each skill into a short learning path, similar to *Chapter 1.6*.
3. Set a realistic timeline, with checkpoints rather than a single distant deadline.
4. Keep the plan in a Project, as in *Chapter 1.11*, so your progress and materials stay in one place.



Watch out: An overloaded learning plan is easy to write and hard to finish. Choose fewer skills and a slower pace rather than a long list you are unlikely to complete.

Complete a demonstrable project

Reading about a skill is not the same as being able to use it. Choose a small project that forces you to apply what you are learning to something real, ideally something close to the actual work of your target role.



Real-world example: Someone learning data analysis to move into an operations role does not stop at tutorials. She analyzes her current team's own data to answer a real question, such as which process step causes the most delay, and produces a short report she can show later.

Test genuine understanding rather than dependence on AI

It is possible to produce good-looking work with heavy help from Claude while understanding very little of it yourself. Guard against this by testing your own understanding directly:

- Explain a concept back in your own words without looking at your notes
- Complete a small task without asking Claude for help, then check your work against what you learned
- Notice when you can follow an explanation but could not have produced it yourself and treat that as a sign to study the topic further



Good practice: Schedule a short, honest self-test at the end of each stage in your learning plan, before moving to the next stage. Catching a weak spot early is far easier than discovering it during your demonstrable project.

Try it now

Research your target role and list its current key skills. Rate your level in each one, choose your top two gaps, and sketch a short learning path for the first gap.

Check the result

- ☐ Did your research include current sources, not only your existing assumptions about the role?
- ☐ Is your learning plan realistic given your available time?
- ☐ Have you planned a way to test your own understanding, separate from what Claude produces for you?

Learning new skills is only part of becoming more valuable at work. *Chapter 4.5* looks at the other half: the human strengths that AI fluency should sharpen, not replace.

Chapter 4.5 Combine AI Fluency with Human Strengths

Skill with Claude is valuable, but it is not the whole of what makes you valuable at work. This chapter looks at the strengths that stay firmly yours, and how to keep them sharp while you automate more of your routine tasks.

Develop human strengths

There are seven areas that depend on human experience, relationships, and context in ways Claude cannot fully replace:

1. **Judgment** means weighing a situation's specifics and deciding what matters most, especially when the right answer is not obvious
2. **Domain knowledge** is the deep, specific understanding of your field that lets you spot when something looks wrong
3. **Communication** is explaining an idea clearly to the particular person or group in front of you
4. **Creativity** is combining ideas in a way that fits your actual situation, not a generic best practice
5. **Empathy** is understanding how a decision will actually land with the people affected by it
6. **Leadership** is helping other people do their best work, especially when the path forward is unclear
7. **Stakeholder management** is understanding what different people need from a project and balancing those needs



Real-world example: Two analysts both use Claude to build the same kind of financial model. One stops there. The other uses her domain knowledge to notice that the model's assumption about seasonal demand does not match what she has seen in three years of actual sales data. Her judgment, not the model itself, is what protects the business from a bad decision.

Recognize skill atrophy

Skill atrophy happens when a skill you rarely practice grows weaker without your noticing, even though you might not notice until you need it. Heavy reliance on Claude for tasks you once did yourself can accelerate this, especially for skills tied to writing, calculation, and structured thinking.

Watch for signs of atrophy: hesitating over a task you used to do easily, or noticing that you can no longer catch an obvious error in Claude's work without help. When you notice this, set aside time to practice the skill without assistance, the way you tested your own understanding in *Chapter 4.4*.



Watch out: Do not let convenience replace a skill you still need without your noticing. A skill you never practice becomes a skill you can no longer trust yourself with, even in situations where Claude is unavailable or wrong.

Retain responsibility for decisions

Claude can draft, analyze, and suggest, but it does not carry consequences the way you do. When a decision affects your work, your team, or another person, the responsibility for that decision stays with you, regardless of how much of the drafting or analysis Claude provided.

This means reviewing important output carefully, questioning a result that seems off even if you cannot immediately explain why, and being ready to explain and defend a decision in your own words.

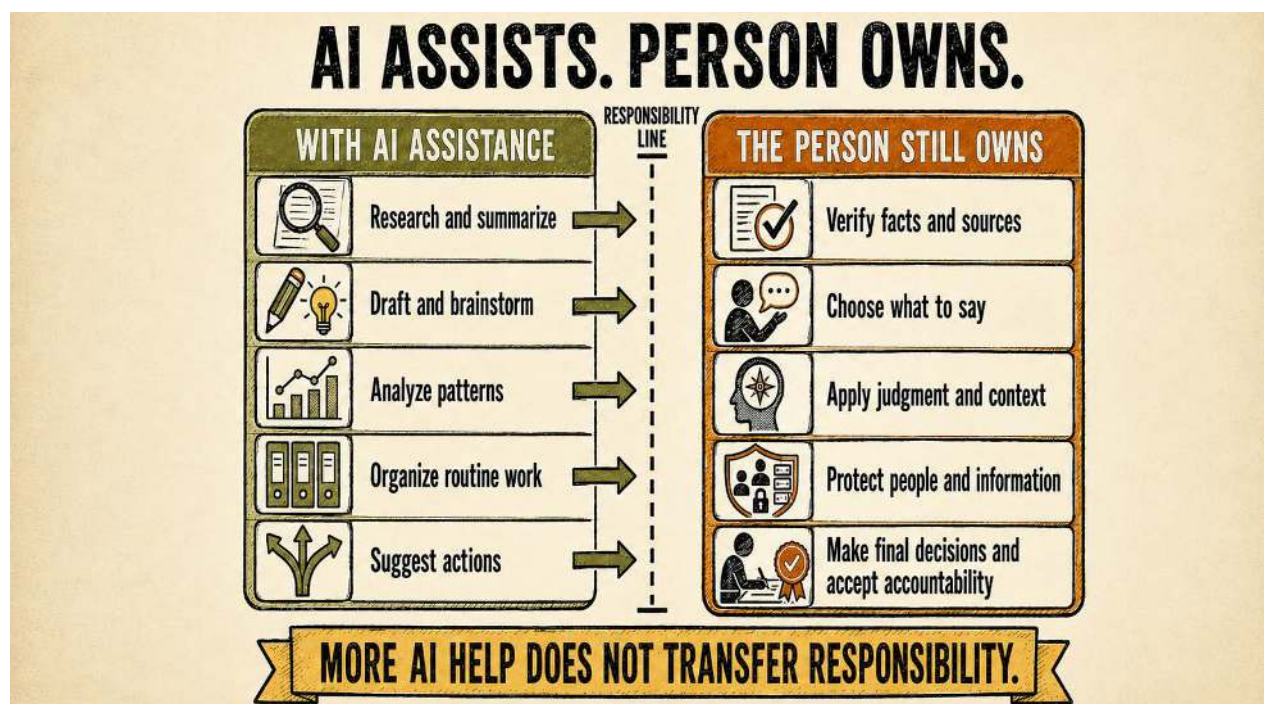


Figure 4.5A: AI can assist with a task, but responsibility for the decision and its outcome stays with the person who made it



Good practice: Before you finalize any consequential piece of work, ask yourself whether you could explain and defend it in a meeting without reading from the screen. If you cannot, review it more closely before you rely on it.

Try it now

Pick one of the seven human strengths listed in this chapter. Write one specific example from the past month where that strength, not Claude, made the difference in a piece of your work.

Check the result

- ☐ Can you name a real example, not a general claim, of a human strength you used recently?
- ☐ Have you noticed any skill of your own growing weaker from disuse?

- ☐ Do you have a clear sense of which decisions in your work remain yours, no matter how much Claude contributes?

Once you know where your own judgment adds the most value, you're ready to bring that combination to your organization. *Chapter 4.6* shows you how to propose and lead an AI improvement project at work.

Chapter 4.6 Propose and Lead an AI Improvement at Work

The improvement you measured in *Chapter 4.3* might be worth more than a personal habit. If it could help your whole team, this chapter shows you how to propose and lead that change responsibly.

Define the business problem

Start from the problem, not the tool. Describe what is currently slow, error-prone, or costly, in terms a colleague outside your immediate team would understand.

A clear business problem sounds like “Our weekly reporting takes six hours and often contains outdated figures,” not “We should use Claude more.”

Estimate benefits and risks

List the likely benefits of your proposed improvement, using the measurements from *Chapter 4.3* where you have them. Then list the risks honestly: what could go wrong, who could be affected, and how serious the consequences would be:

- Benefits might include time saved, fewer errors, or faster turnaround for colleagues who depend on the work
- Risks might include exposing sensitive data, producing a result nobody checks carefully, or creating a dependency on a tool that changes without notice



Watch out: A proposal that only lists benefits will look less credible, not more. Naming the real risks, and how you plan to manage them, shows that you have thought the idea through.

Follow organizational policy

Before proposing anything, check your organization's AI policy, covering approved tools, data handling rules, and required approvals. *Chapter 2.1* introduced these rules for your own work. A team-wide proposal needs the same care, applied more broadly.

If your organization does not yet have a clear policy for this kind of change, say so in your proposal, and suggest who should be involved in setting one.

Win stakeholder support

Identify who needs to approve or support your proposal, and what each of them cares about most. A manager may care about cost and risk. A colleague who will use the new process may care about how much it changes their daily work:

1. List the people whose support you need.
2. Note what matters most to each person.
3. Prepare a short explanation for each stakeholder, focused on what matters to them.

4. Ask for feedback before asking for final approval.

ONE-PAGE PILOT CANVAS

1. THE PROBLEM



2. THE EVIDENCE SUPPORTING IT



3. THE CONTROLS THAT MANAGE RISK



4. HOW SUCCESS WILL BE MEASURED



5. THE APPROVAL NEEDED TO PROCEED



APPROVED ☐

SIGNATURE _____

DATE _____

Figure 4.6A: A one-page pilot canvas puts the problem, the evidence, the controls, the measurement plan, and the approval step on one page

Run a limited pilot

Do not propose a full rollout as your first step. Propose a small, time-limited pilot with a clear group of participants, a clear duration, and a clear way to measure the result:

1. Choose a small group or a short period for the pilot.
2. Set the same kind of baseline and measurements from *Chapter 4.3*.
3. Run the pilot and collect feedback from participants as it proceeds.
4. Set a specific end date to review the results.



Good practice: Build a stop condition into the pilot from the start: a clear sign that would tell you to pause or cancel it. A pilot with no way to stop safely is harder to trust, both for you and for the people approving it.

Report results honestly

When the pilot ends, report what actually happened, including any part that did not go as planned. Use your measurements from *Chapter 4.3* to show the real change, not an estimate.

If the pilot succeeded, propose a clear next step, such as expanding to a larger group. If it did not, explain what you learned and whether the idea is worth trying again in a different form.

Try it now

Draft a one-page pilot canvas for your *Chapter 4.3* improvement, or for a new idea worth proposing to your team. Include the problem, the evidence, the risks and controls, and how you would measure success.

Check the result

- ☐ Does your proposal name a specific business problem, rather than starting from the tool?
- ☐ Have you checked your organization's current AI policy before proposing this change?
- ☐ Does your pilot have a clear size, duration, and stop condition?

Leading a successful pilot project gives you a concrete result to point to. *Chapter 4.7* shows you how to turn that result, along with the rest of your achievement record, into a case for promotion or higher pay.

Chapter 4.7 Build a Case for Promotion or Higher Pay

If *Chapter 4.1* pointed you toward promotion or higher pay within your current organization, this chapter helps you turn your evidence into a case that others will find convincing.

Maintain an achievement record

Chapter 4.3 introduced the achievement record: a running list of your measured improvements. This is the raw material for your case, so keep adding to it well before you plan to ask for anything.

A thin record built the week before a performance conversation is far less convincing than a steady one built over months.

Connect work to organizational outcomes

A manager rarely cares about a task getting faster for its own sake. They care about what that speed, quality, or reliability means for the organization: money saved, risk reduced, customers served better, or a project delivered on time.

For each entry in your achievement record, add one line connecting it to an outcome your organization already cares about. "Cut report preparation from six hours to two" becomes stronger as "Cut report preparation from six hours to two, freeing time for the client research that led to two renewed contracts."

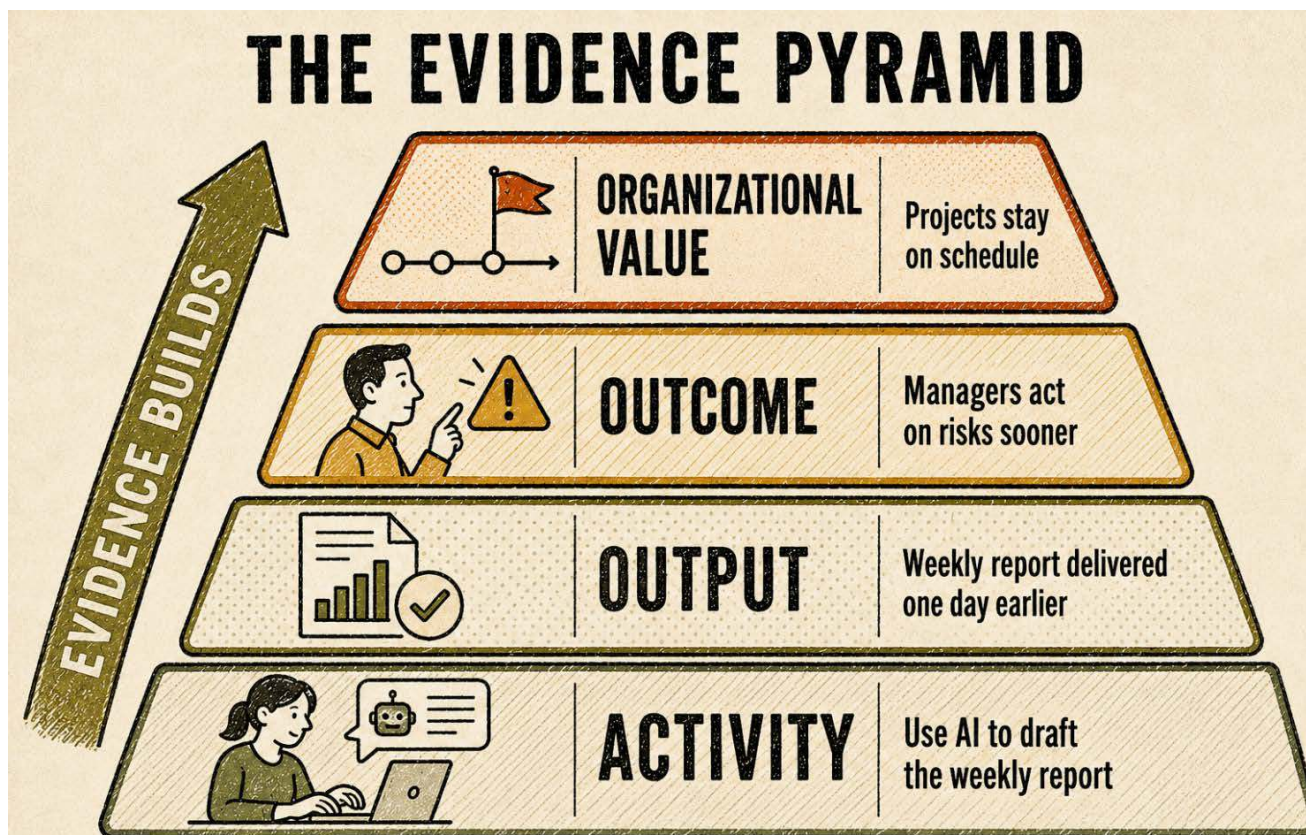


Figure 4.7A: An evidence pyramid moves from what you did, to what you produced, to what changed, to what it meant for the organization

Describe expanded responsibilities

A case for promotion often depends on showing that your actual responsibilities have grown, not only that your existing tasks have improved. Look for ways your role has genuinely expanded: new kinds of work you now handle, decisions you now make without being asked, or colleagues who now come to you for help.



Watch out: Do not claim responsibility for work you only touched briefly or supported in a minor way. An overstated claim is easy to challenge and can undermine the credibility of your entire case.

Prepare for a performance conversation

Before a formal review or a direct request, prepare as you would for any important conversation:

1. Choose your three or four strongest pieces of evidence from your achievement record.
2. Practice explaining each one in two or three sentences, focused on the outcome.
3. Anticipate likely questions or objections and prepare honest answers.
4. Decide what you are specifically asking for, whether that is a new title, a raise, or expanded responsibilities.



Real-world example: A customer support lead preparing for her annual review does not walk in with a general sense that she “did a lot with AI this year.” She brings three specific examples, each with a before-and-after measurement and a connection to a result her manager already tracks, such as customer response time.

Propose the next contribution

Beyond looking back at what you have done, describe what you would do with expanded responsibility or a new role. A specific, credible plan for the future often matters as much as a strong record of the past.

Case for Promotion — One-Page Summary

Jordan Ellis • Project Coordinator • Prepared for annual review, July 2026

1. Weekly status report	
Activity	Automated the weekly status report using a saved Claude prompt fed by pasted source data.
Output	Cut report preparation from 90 minutes to 35 minutes each week.
Outcome	Reports now go out consistently by Friday morning instead of trickling in over the weekend.
Value	Frees roughly 4 hours a month for client <u>research</u>, and gives leadership more reliable visibility into project status.

Figure 4.7B: A one-page summary turns your achievement record into a case built from activity, output, outcome, and organizational value



Sample files: A one-page Word summary for a case for promotion: <https://github.com/markjprice/claude-vb/tree/main/Files/Book4/4-7-Case-for-Promotion.docx>. Three achievement entries reuse the same tasks from *Figure 4.3B* (status report, onboarding emails, meeting notes) for continuity across the book.



Good practice: Rehearse your case out loud before the actual conversation, ideally with a colleague who can ask you questions. Hearing your own case spoken aloud often reveals gaps that are easy to miss on the page.

Try it now

Choose your three strongest achievement-record entries. For each one, write a line connecting it to an organizational outcome, then write one sentence proposing what you would take on next.

Check the result

- ☐ Does each piece of evidence connect clearly to something your organization already values?
- ☐ Have you avoided overstating your role in any piece of evidence?
- ☐ Do you have a specific ask, rather than a general hope that good work will be noticed?

A strong case for promotion works inside your current organization. *Chapter 4.8* prepares you for a move outside it, if that's the direction you choose, covering how to research a new role and build a résumé you can stand behind.

Chapter 4.8 Prepare for a Better Job or Career Change

If *Chapter 4.1* pointed you toward a career change, this chapter takes you from research through to a completed application, using the skills you built earlier in this book.

Research resilient and growing roles

Before committing to a target role, research whether it is a good bet for the years ahead. Look for roles with steady or growing demand and be honest about how automation and AI might affect that demand over time:

1. Ask Claude to search for current data on demand and outlook for your target role.
2. Check at least one source directly, using the research habits from *Chapter 1.8*.
3. Note anything the sources disagree about, rather than picking the most convenient answer.
4. Consider roles adjacent to your first choice, in case they offer a stronger outlook.

Identify skill gaps

Return to the current-to-target skills map you built in *Chapter 4.4* or build one now if you have not already. A career change usually needs a slightly wider set of skills than a promotion within your current field, so review the map with fresh eyes.

Improve a résumé and LinkedIn profile honestly

Claude can help you write a clearer résumé and profile, but the content must remain accurate. Every claim should describe something you actually did:

1. Provide Claude with your work history and the target role's requirements.
2. Ask for a draft that highlights relevant experience clearly.
3. Check every claim against what you actually did, and correct anything overstated.
4. Ask for a second pass focused on clarity and length, not new claims.



Watch out: An impressive-sounding résumé built on exaggerated claims can fall apart in an interview, when someone asks a specific question about a project you described. Keep every claim something you can defend in detail.

Prepare work samples

If your target role values demonstrated skill, prepare one or two samples that show your ability directly, rather than only describing it. *Chapter 4.9* covers how to build a full portfolio case study, but even a single strong sample helps here.



Real-world example: Someone moving from general marketing into data-informed marketing does not just add “data skills” to a résumé. She includes a short writeup of one campaign where she used data to change the approach midway through, with the result that followed.

Practice interviews

Use Claude to prepare for likely interview questions, based on the target role's requirements and common questions in that field:

1. Ask Claude to generate likely interview questions for the role.
2. Answer each question out loud, then review your answer against the question's intent.
3. Ask Claude to act as an interviewer and ask follow-up questions based on your answers.
4. Practice explaining your work samples in two or three sentences each.

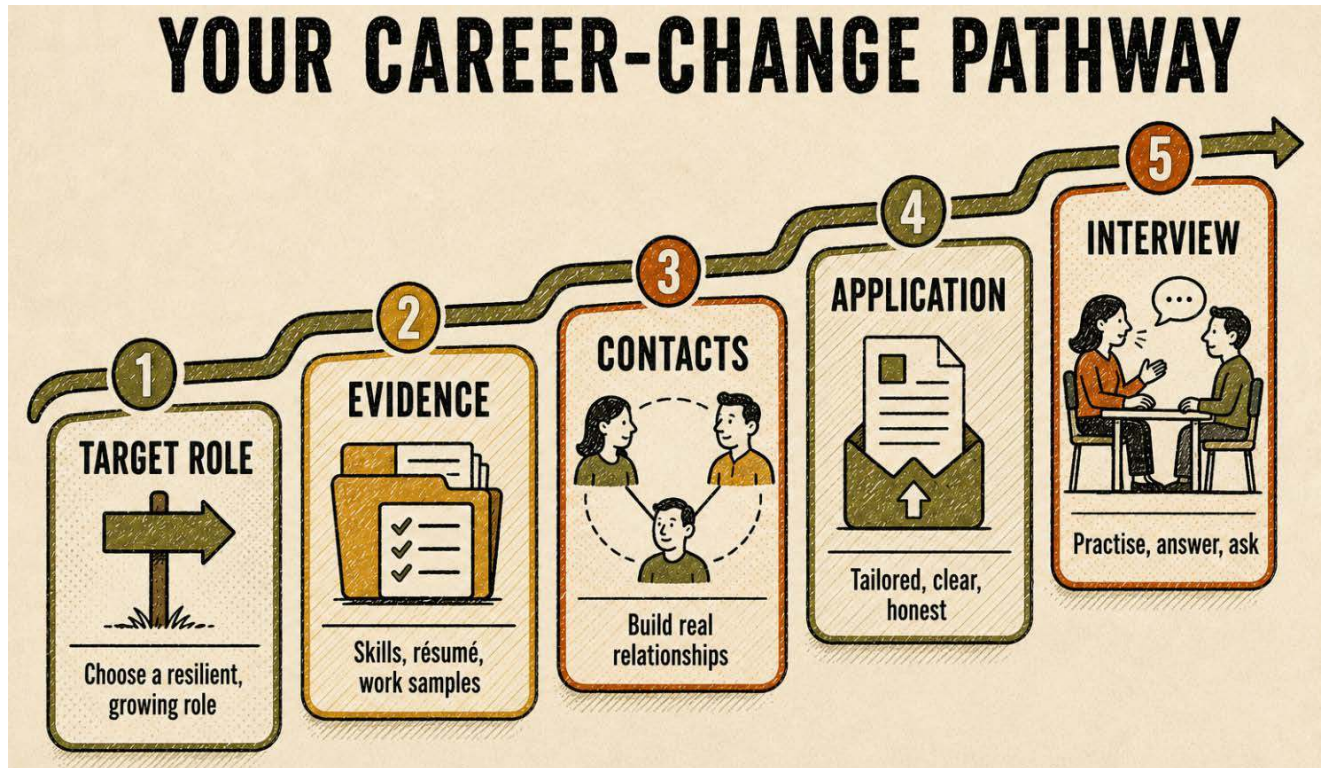


Figure 4.8A: A career-change pathway moves from a target role, through evidence and contacts, to an application and an interview

Build a networking plan

Most career changes benefit from real conversations with people already in or near the target role, not only applications sent into a system. Use Claude to help you prepare for these conversations, not to replace them:

1. List people you already know who work in or near the target field.
2. Draft a short, specific message asking for a brief conversation, not a job.
3. Prepare two or three questions that show genuine interest in their experience.
4. Follow up with a thank-you message that references something specific from the conversation.



Good practice: Treat every networking conversation as a chance to learn something real about the role, not only as a step toward an opportunity. People notice the difference, and genuine curiosity tends to lead to better conversations.

Try it now

Research your target role's outlook, then draft one paragraph of your résumé aimed at that role. Check every claim in the paragraph against your actual experience before moving on.

Check the result

- ☐ Is your target role's outlook based on current research, not an old assumption?
- ☐ Does every claim in your draft résumé describe something you could explain in detail if asked?
- ☐ Do you have at least one real person you could reach out to for a networking conversation?

A résumé states your experience in a few lines. *Chapter 4.9* shows you how to back it up with a full portfolio and a measurable case study that lets someone else see your work in detail.

Chapter 4.9 Build a Portfolio and Measurable Case Study

A case study is a short, honest account of a real problem you improved, with evidence to prove it. This chapter shows you how to build one, whether it will support a promotion, a career change, or an independent offer.

Select an internal, personal, or client project

Choose one project that shows your ability clearly, drawing from your achievement record in *Chapter 4.3*. It can come from your current job, a personal project, or client work, as long as you have the right to share the details.

Pick a project with a clear before and after, rather than one with vague or gradual improvement. A reader should be able to see the change quickly.

Explain the original problem

Describe the problem the way it looked before you addressed it: what was slow, costly, unreliable, or frustrating, and for whom. A reader unfamiliar with your work should understand the problem within a sentence or two.

Show the process and safeguards

Explain what you actually did, at a level of detail that builds trust without becoming a full tutorial. Include the safeguards you used, such as verification steps, human review points, or the acceptance criteria from *Chapter 3.3*, since these show a careful approach rather than a rushed one.



Watch out: A case study that only shows the polished result can raise doubts about how the work was actually done. Naming your checks and safeguards makes the achievement more credible, not less impressive.

Demonstrate improvement

Present your measured result clearly, using the same before-and-after approach from *Chapter 4.3*. A specific number, chart, or comparison is far more convincing than a general claim that something improved.

This step covers the last two parts of the one-page structure: the result you achieved, and the evidence behind it. State them together, since a reader should never have to take a number on trust when a quick check is available:

- **Result:** a single sentence naming the before number and the after number side by side, such as “invoice processing time fell from twelve minutes to four”
- **Evidence:** the artifact that lets someone else verify the result without relying on your word alone, such as a screenshot, an exported report, a dashboard view, or a short quote from someone affected by the change

Choose the measure your reader already cares about, using the same time, quality, errors, or outcome categories from *Chapter 4.3*, rather than whichever number happens to look most impressive.

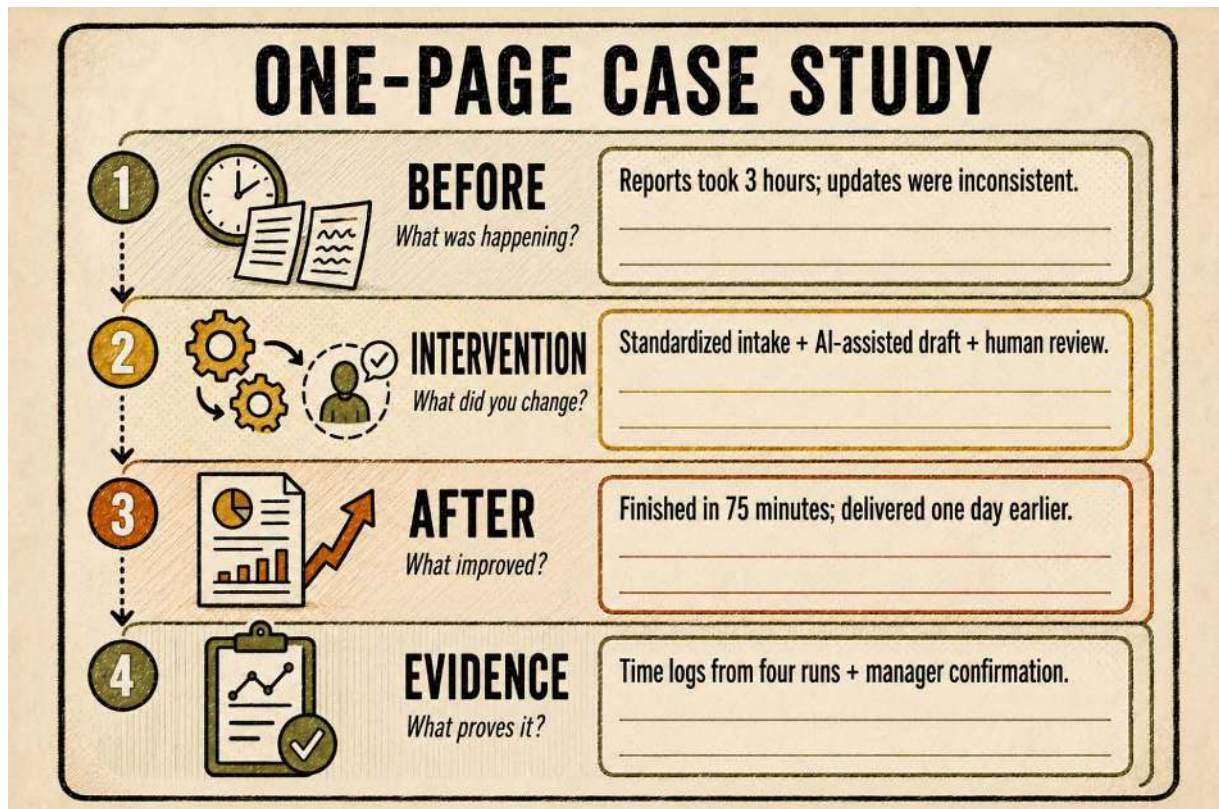


Figure 4.9A: A one-page case study moves from the original problem, through what you did, to the result and the evidence behind it

Remove confidential information

Before sharing a case study with anyone outside your organization, or even with certain people inside it, review it for confidential material:

1. Remove client names, internal figures, and proprietary details unless you have explicit permission to share them.
2. Replace specific numbers with proportional ones where necessary, such as “cut processing time by 40 percent” instead of exact minutes tied to a specific system.
3. Ask a colleague or manager to review the case study if you are at all unsure what is safe to share.
4. Keep an unredacted internal version separately, for your own records and for conversations where full detail is appropriate.



Real-world example: An operations analyst wants to share a case study about a workflow that cut invoice-processing errors. Before sharing it publicly, she replaces the client's name with "a mid-sized logistics client" and reports the improvement as a percentage rather than exact dollar figures tied to a specific contract.

Case Study: Cutting Weekly Reporting Time Without Losing Accuracy

A mid-sized services team's weekly status report, before and after a Claude-assisted workflow.

Weekly reporting dropped from 3 hours to 75 minutes, with reports delivered a day earlier and no drop in accuracy.

1	BEFORE <i>What was happening?</i> The weekly status report took about three hours to complete, most of it spent gathering figures from three separate systems and reformatting them by hand. Updates were inconsistent week to week, since the result depended on whoever happened to write that week's report.
2	INTERVENTION <i>What did you change?</i> Standardized the intake so the same data points are pulled the same way every week. Built a Claude prompt that

Figure 4.9B: A finished case study fits on one page and moves cleanly from the original problem to the proven result



Sample files: A one-page Word case study: <https://github.com/markjprice/claude-vb/tree/main/Files/Book4/4-9-Case-Study.docx>. This mirrors *Figure 4.9A* exactly using the same weekly-reporting scenario.



Good practice: Write your case study for a reader who does not know you and has thirty seconds to decide whether to keep reading. A clear headline claim, followed by the evidence, works better than a long, gradual buildup.

Try it now

Choose one project from your achievement record. Draft a one-page case study using the before, intervention, after, and evidence structure, and review it once for anything confidential that needs removing.

Check the result

- ☐ Would a reader unfamiliar with your work understand the original problem within a sentence or two?
- ☐ Does your evidence include a specific measurement, not only a general claim?
- ☐ Have you checked the case study for confidential information before sharing it outside your organization?

A strong case study can support a job application, or something more independent. *Chapter 4.10* shows you how to turn a problem you understand well into a specific offer or product you could sell.

Chapter 4.10 Turn a Problem into a Service or Product

If *Chapter 4.1* pointed you toward earning independently, this chapter is where that idea becomes a specific service or product. The best service or product ideas start with a real problem someone else has, not with a skill you want to sell.

Select a customer group

Choose a specific group of people or businesses who share a problem you understand well, ideally one close to your own experience or professional background. A narrow, well-understood group is easier to serve well than a broad, vague one.

“Small accounting firms that struggle to keep client reports consistent” is easier to design an offer around than “small businesses.”

Validate the need

Before building anything, confirm that your chosen group actually experiences this problem, and cares enough to want it solved. Talk to real people in the group if you can, and use Claude to help you research how the problem is discussed publicly, such as in forums or reviews.



Watch out: It is easy to convince yourself a problem is common because you personally find it frustrating. Confirm the need with people outside your own perspective before investing further time.

Research alternatives

Look at how people in your target group currently solve, or try to solve, this problem. This might include competing services, do-it-yourself approaches, or living with the problem unresolved:

1. Search for existing services or products aimed at the same problem.
2. Note their strengths and their gaps, especially anything customers complain about.
3. Identify what you could do differently or better, based on your own experience.
4. Confirm your planned offer is not a repeat of something already well served.

Define outcome, scope, exclusions, and evidence

Write a clear description of what your offer actually delivers:

- **Outcome:** the specific result the customer gets, described in terms that matter to them
- **Scope:** exactly what is included, stated plainly enough to avoid confusion later
- **Exclusions:** what is not included, so expectations stay realistic on both sides
- **Evidence:** proof that you can deliver this outcome, such as the case study from *Chapter 4.9*

Choose service or reusable product

Decide whether your offer works best as a service you deliver personally for each customer, or as a reusable product built once and used by many customers.

A service usually starts faster and depends more on your direct time. A reusable product takes longer to build well but can serve more customers without growing your time commitment in the same way. Many independent workers start with a service and turn a proven, repeated version of it into a product later.

OFFER CANVAS

1 CUSTOMER
Who is it for?

2 PROBLEM
What needs fixing?

3 OUTCOME
What will improve?

4 PROOF
Why believe you?

5 SCOPE
What is included?

6 PRICE
What will it cost?

7 RISK
What makes it safe to try?

Figure 4.10A: An offer canvas combines the customer, the problem, the outcome, your proof, the scope, the price, and the risk in one place

Price and package responsibly

Set a price that reflects the value delivered and the effort required, and be honest with yourself about your actual costs, including your time. Avoid pricing based only on what feels comfortable to charge, since that often leads to underpricing real value:

1. Estimate your actual cost, including time, at a rate you would be comfortable defending.
2. Research what comparable offers charge in your market.
3. Set a price and clearly state what it includes.
4. Decide how you will handle requests that fall outside your defined scope.



Good practice: Write your price and scope down before your first real conversation with a prospective customer. Deciding in the moment, under pressure to close a deal, tends to produce inconsistent and sometimes regrettable terms.

Try it now

Draft an offer canvas for one problem you could realistically help solve. Fill in the customer, the problem, the outcome, your proof, the scope, a price, and the main risk.

An optional online-only chapter covers how to find customers and deliver products and services to them responsibly: <https://github.com/markjprice/claude-vb/blob/main/docs/ch4-12-find-customers.md>.

Check the result

- ☐ Have you confirmed the need with someone outside your own perspective?
- ☐ Does your scope clearly state what is included and what is not?
- ☐ Is your price based on real cost and market research, rather than a guess?

This chapter defined your offer. You’ve now covered every path this book offers: improving your current role, preparing for a new one, or building something to offer customers directly. *Chapter 4.11* asks you to choose one path and commit to a real 30-day project, measured against a baseline you set yourself.

Chapter 4.11 Project – Complete a 30-day Career Value Project

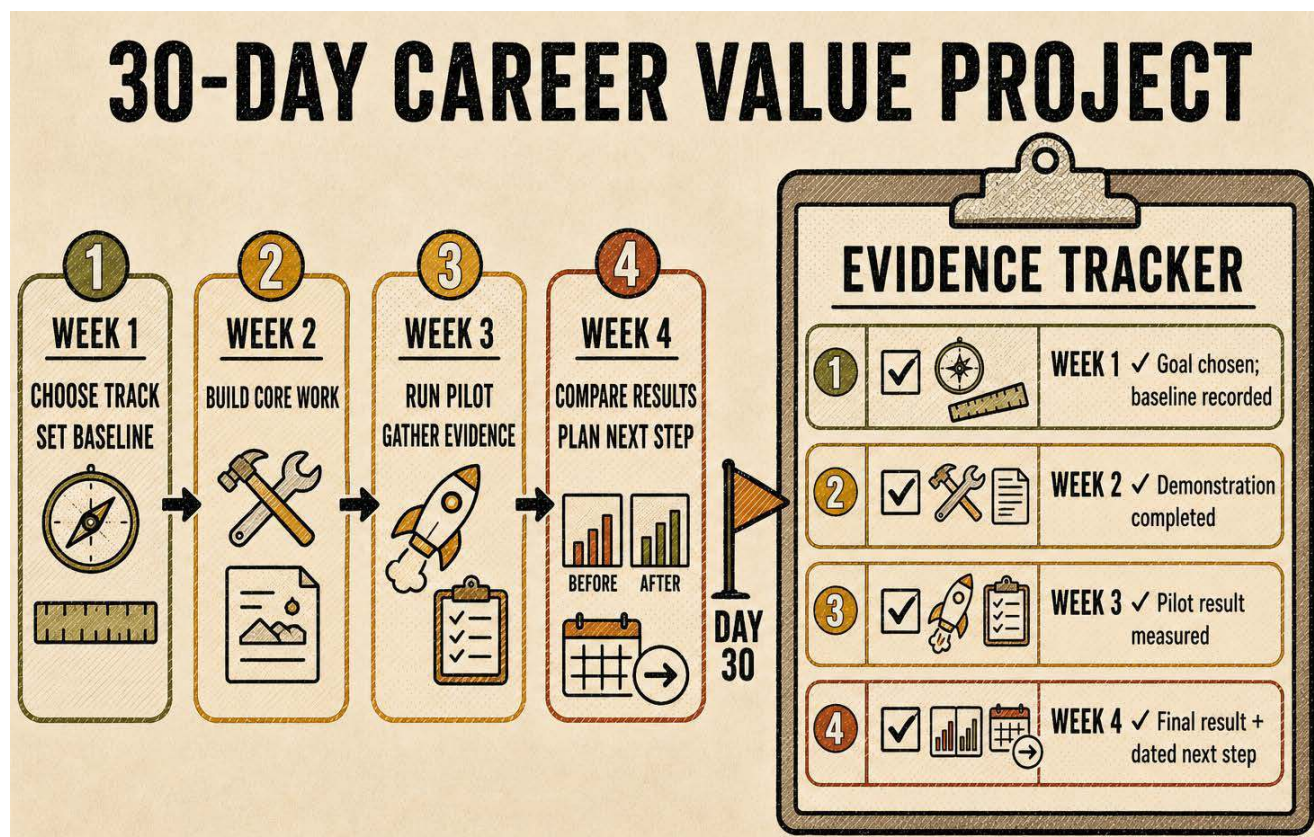
This final project brings together the whole book: choosing a direction, finding high-value work, measuring your improvement, and building a case for what comes next. Over the next thirty days, you will complete one track fully, with real evidence to show for it.

Choose your track

Pick the track that matches the direction you chose in *Chapter 4.1*:

1. **Improve your present job:** create and run one approved workflow pilot, compare it with your baseline, and prepare a one-page results report.
2. **Prepare for promotion or career change:** produce a skill-gap plan, a completed demonstration, a portfolio case study, and a conversation or application plan.
3. **Earn independent income:** validate one offer, produce a demonstration, approach a defined number of suitable prospects, and complete or arrange one pilot.

You may recognize pieces of your track from earlier chapters in this book. This project asks you to complete the whole sequence, not only the individual steps.





Watch out: Thirty days is enough time to make real progress, not enough time to make everything perfect. Set a defined scope for your pilot before you start, and resist the urge to expand it once you are underway.

Review results

At the end of the thirty days, compare your result against your baseline, using the same honest approach from *Chapter 4.3*:

1. **Baseline:** record the task as you do it before AI enhancement.
2. **Weekly updates:** record the task with AI enhancements.
3. **Result:** calculate the time saved as an absolute and as a percentage reduction.



Sample files: A spreadsheet launch tracker: <https://github.com/markjprice/claude-vb/tree/main/Files/Book4/4-11-Launch-Tracker.xlsx>. It continues the same weekly-status-report thread from *Figure 4.3B* and *Figure 4.9B*.

	A	B	C	D	E	F
1	30-Day Career Value Project — Launch Tracker					
2	<i>Track: Improve your present job — automate the weekly status report</i>					
3						
4		Stage	Week / Date	Time (min)	Notes / Evidence	
5	①	Baseline	Week 0 — 28 Jun	90	Manual report, gathered by hand from three separate systems.	
6		Week 1	05 Jul	70	Dry run of a Claude-assisted draft prompt; no live report sent yet.	
7		Week 2	12 Jul	55	First live report using the new workflow; refined the prompt based on gaps found.	
8	②	Week 3	19 Jul	45	Second live cycle; added a human review checkpoint before sending.	
9		Week 4	25 Jul	35	Third and fourth live cycles; time held steady with zero formatting errors.	
10	③	Final Result	Week 4 wrap — 25 Jul	35	Time logs from four weekly cycles, confirmed by the reporting manager.	

Figure 4.11B: A completed launch tracker records your baseline, your weekly progress, and your final result in one place. Some readers may build this as a Claude Artifact instead of a document or spreadsheet

Produce a next-step plan

Regardless of how the thirty days went, write a short plan for what comes next. If the pilot succeeded, describe how you will expand or repeat it. If it fell short, describe what you learned and what you would try differently:

1. Summarize what you set out to do and what actually happened.
2. State whether you met the goal you set at the start.
3. Describe the single most important lesson from the thirty days.

4. Write your next concrete step, with a realistic date.



Good practice: Share your results with at least one other person, whether a manager, a mentor, or a peer. Explaining your project out loud often clarifies what you have actually learned, and an outside perspective can catch something you missed.

Try it now

Choose your track, set your baseline this week, and block time on your calendar for each of the four weeks ahead. Begin the first stage of your pilot.

Check the result

- ☐ Did you complete the full sequence for your track, not only part of it?
- ☐ Does your final result compare clearly against the baseline you set at the start?
- ☐ Do you have a specific, dated next step written down?

Wrap up with the Epilogue

Whichever track you chose for your thirty-day project, you've now used Claude to ask better questions, create real work, automate a genuine routine, and advance something that matters in your career or income.

The *Epilogue* looks back across all four books and points to where you might go next, especially if the automations and workflows you built have left you curious about writing and running code yourself.

Epilogue

In this epilogue, you will learn:

- Where this book ends and your work continues
- What stays true as Claude changes
- Why you should spend your next \$20 on a second AI
- What to do if you want to build, not just direct
- A last word

Where this book ends and your work continues

You started this volume by asking Claude a question. You ended it with improving your current job tasks, or building a documented case for a promotion, or a real product or service offering to a paying customer. That progress did not come from one clever trick. It came from four stages, each building on the one before it.

Book 1 taught you to ask Claude better questions: how to set up your account, hold a real conversation, and use Claude to learn, research, and decide. You finished it with a personal AI workbench built around your own life and work.

Book 2 turned those conversations into finished results. You drafted and revised documents, made sense of spreadsheets and long reports, created visual and interactive content and presentations, and connected Claude to real files and workplace information. You finished it by building a useful weekly briefing that serves a real audience.

Book 3 showed you how to stop repeating yourself. You started by turning your weekly briefing into a recurring task that runs on a schedule, converted a manual routine into a skill, and connected trusted tools with MCP. You finished it with a complete workflow, tested against a baseline and checked to see what happens when something goes wrong.

Book 4 asked what all of that was for. You measured the value of your improved work, learned the skills your next role requires, and built a case for a promotion, a new job, or a service of your own. You finished it with a thirty-day project measured against a result you set for yourself, not a guess.

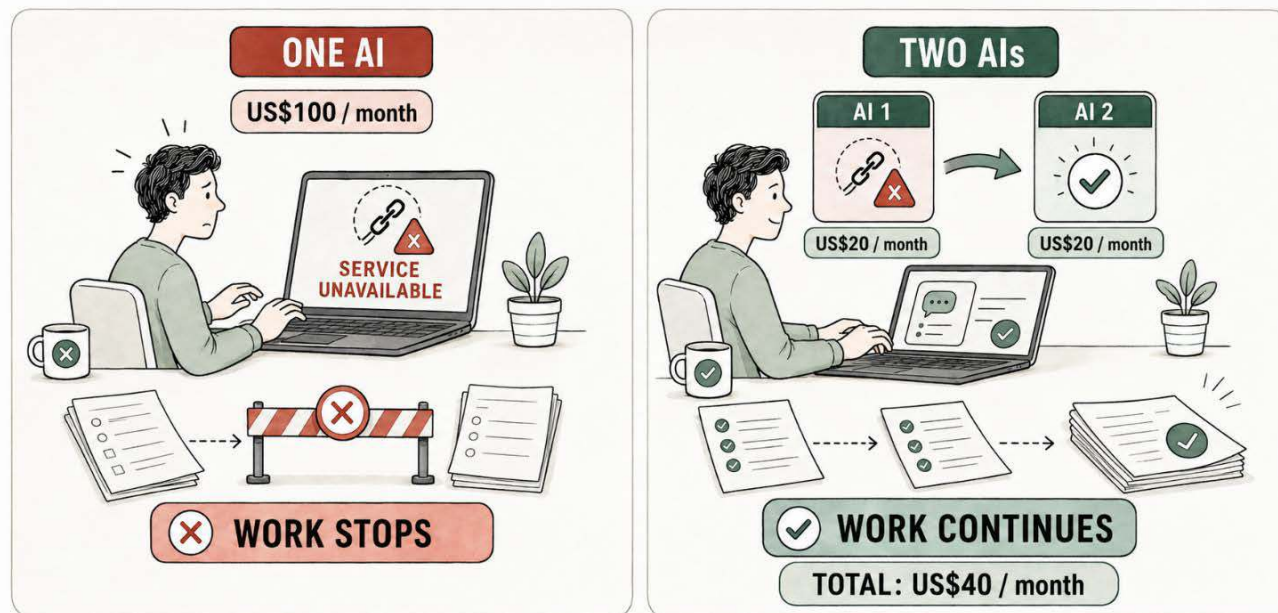
What stays true as Claude changes

Claude will keep changing. Menu names, default modes, and exact controls may look different by the time you read this. Some of the screenshots in these books are probably already a step behind the current interface.

What should not change is the way you approached each task: define what you want, check your employer's policy, verify what Claude gives you, and decide for yourself whether the result is good enough to use. That judgment is the real skill. It will transfer to whatever Claude looks like next year, and to any AI tool you use after that.

Why you should spend your next \$20 on a second AI

When Claude starts running into its limits, the obvious next step is to pay Anthropic more: upgrading from Pro to Max, from around US\$20 a month to US\$100 or US\$200, depending on the tier available in your country at the time. Before you make that jump, consider a different one. Keep your Pro plan at US\$20 and spend another US\$20 on a second AI tool instead.



Epilogue Figure 1: Using two AI services instead of one larger plan provides redundancy as well as a second perspective

Think of Claude and ChatGPT, from OpenAI, as the Coke and Pepsi of AI assistants. Both are genuinely capable general-purpose tools, and picking a primary between them is closer to personal taste than a clear technical verdict. Whichever one you did not choose as your primary is usually the strongest possible secondary: not a downgrade, but a second frontier-grade model with its own way of writing, reasoning, and handling a task.

A second full tool earns its US\$20 in a way a bigger single-vendor plan often doesn't. This volume has asked you, in almost every chapter, to check a result rather than trust it on the first try. A second AI extends that same habit: run a demanding request past both, and you get a genuine second opinion instead of the same model reasoning with itself twice. It also protects you from a single company's outage, a single model's blind spot, or losing access to work stored only with one vendor.

That last risk is the easiest to underestimate until it happens to you. Major AI providers can suspend or terminate accounts, sometimes for a genuine policy violation and sometimes because an automated fraud or abuse system flags something ordinary as suspicious, and the reason given is rarely detailed enough to argue with.

Anthropic publishes account-enforcement statistics every six months. Its January–June 2026 report lists 11.4 million banned accounts, about 398,000 appeals, and about 42,000 appeal overturns. Those figures change substantially between reporting periods, so check Anthropic's current Transparency Hub rather than relying on the numbers printed here: <https://www.anthropic.com/transparency/system-trust-reporting>.

OpenAI describes its appeals process for banned ChatGPT accounts but doesn't publish how often it bans accounts or how often appeals succeed.

Losing access can still be disruptive even when the suspension is later overturned. If your work matters, keep important outputs and instructions somewhere you control rather than leaving your only copy inside one AI account.

This is the brutal version of the case made earlier in this section for a second AI subscription: not just a second opinion, but a working account you can still open the day your primary one goes dark. Export or back up anything you would genuinely miss, such as a finished document, a useful custom instruction, or a project you rely on, the same way you would back up a hard drive, rather than trusting your only copy to an account you do not ultimately control.

With Claude as your primary AI service, if you choose ChatGPT as your second AI and you want to learn ChatGPT properly rather than just hope that all its features work in the same way that Claude does, the companion *ChatGPT Visual Bible* walks through it with the same four-stage structure as this volume.

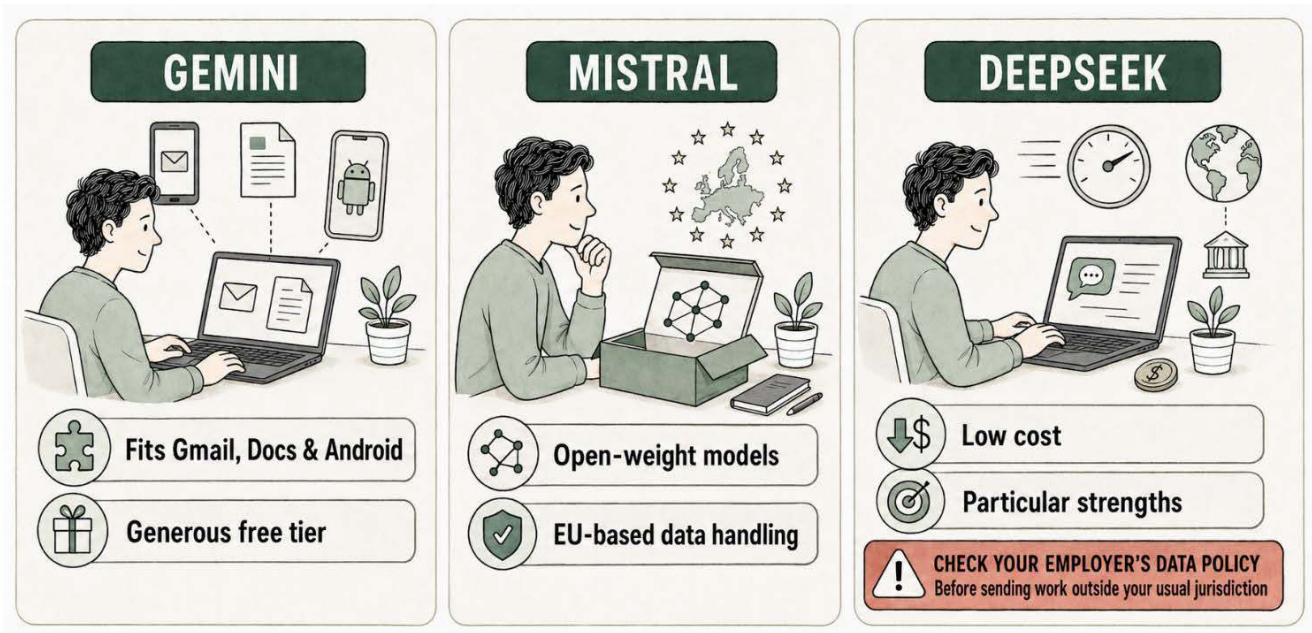
Adding a third AI service

A third AI is even more optional and worth adding only once your first two are earning their keep.

Google's Gemini is the natural choice if you already live inside Gmail, Docs, or Android, and its free tier is generous, so you can try it alongside your paid subscriptions without cost.

Mistral is worth a look if open-weight models or EU-based data handling matter to your work.

And one of the fast-improving Chinese providers, such as DeepSeek, Qwen, or Kimi, can be worth a try for their low cost and particular strengths with coding, though check your employer's data policy, the same way you checked it in *Chapter 2.1*, before you send anything work-related through a platform hosted outside your usual jurisdiction.



Epilogue Figure 2: Choosing a third AI provider

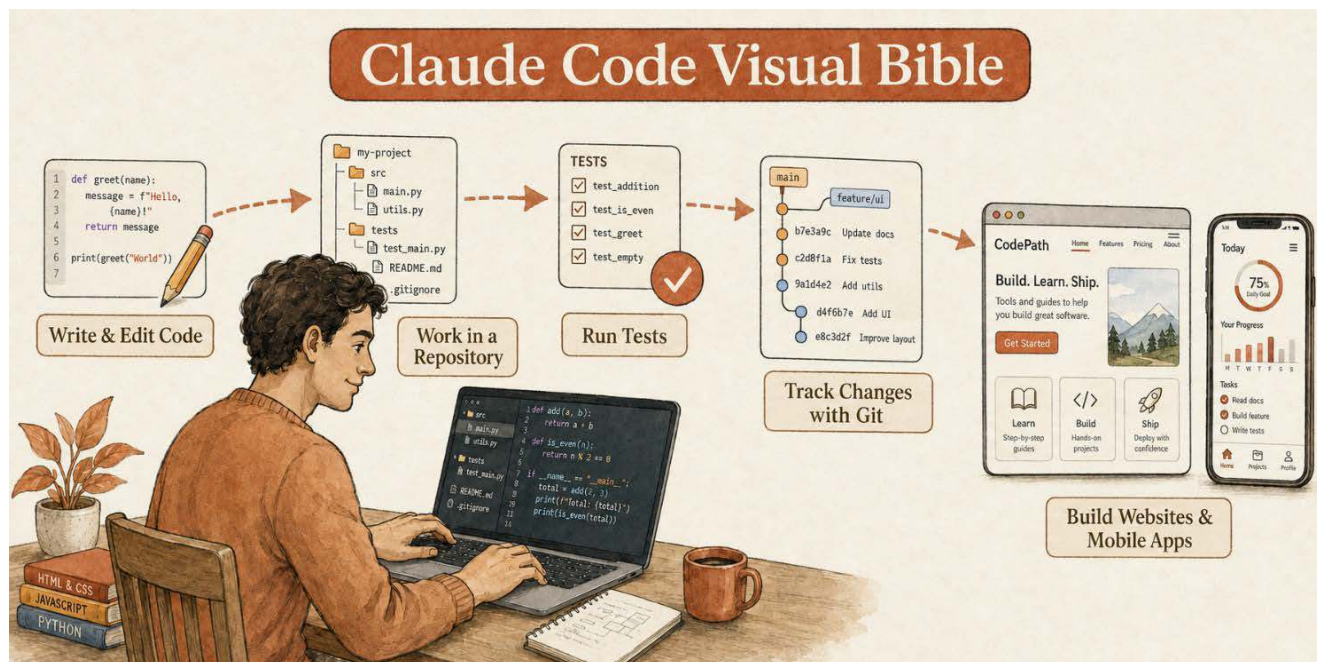
Learn more online: I have also written three bonus books, which you can find online:

- *Beyond Your First AI* (<https://github.com/markjprice/claude-vb/blob/main/docs/bonus/beyond/README.md>): This bonus book explains when a different AI service is the better tool for a specific job, or a backup when your primary AI is unavailable. It covers OpenAI ChatGPT, Google Gemini, Mistral Vibe, and some Chinese providers like DeepSeek and Qwen.
- *Profession-Specific Prompts* (<https://github.com/markjprice/claude-vb/blob/main/docs/bonus/professions/README.md>): This bonus book adds a layer that the general prompting skills taught in the print books do not cover: the specific duties, risks, and rules that come with 20 professions.
- *Local AI Models* (<https://github.com/markjprice/claude-vb/blob/main/docs/bonus/local-models/README.md>): The print books taught you to work with Claude running in Anthropic's cloud. This bonus book covers running an AI model entirely on your own laptop or desktop, with nothing leaving your device. Since Anthropic does not release any of its own models, this uses free and open-weight models like Google Gemma 4, Alibaba Qwen3.8, Meta Muse Glimmer, and NVIDIA Nemotron 3.5 Lightning.



What to do if you want to build, not just direct

In *Book 3*, you built automations that pushed right up against the edge of this book's scope. You customized skills, connected tools with MCP, and maybe wondered what it would take to write the code behind an automation yourself, instead of only directing Claude to configure one. That is the boundary between this volume and the next one in the series.



Epilogue Figure 3: Claude Code Visual Bible topics

Claude Code Visual Bible picks up exactly where the automation chapters left off. It teaches you to write and edit real code, work in a repository, run tests, and build websites and mobile apps. It assumes the same starting point as this volume: no coding background, ordinary computer skills, and a willingness to follow clear, visual steps.

You do not need to be planning a career in software to get value from it. Many readers who reach this point just want to understand what their automations are actually doing under the hood, or to build one small tool that solves a problem no existing skill or plugin quite covers. If that describes you, *Claude Code Visual Bible* is the next book in this series worth picking up.

A last word

Four books and forty chapters is a lot of material to work through, especially if you diligently completed all the **Try it now** sections, and you likely skipped some sections that did not apply to you yet. Keep this volume where you can reach it. The chapter you skimmed on MCP or scheduled tasks may be exactly the one you need in six months.

This book was written to close the gap between using Claude and getting real value from it. What you do with that next is up to you. Every chapter in this book asked you to check the result before you trusted it. That habit is the real skill. Everything else was Claude.

Go use it.

Share your thoughts

Now you've finished *Claude Visual Bible*, we'd love to hear your thoughts! Scan the QR code below to go straight to the Amazon review page for this book and share your feedback or leave a review on the site that you purchased it from.



<https://packt.link/r/1808658396>

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

