



A FIELD GUIDE • 2026

FOR PEOPLE WHO RUN AND WORK ON AI

The Claude Handbook

Everything you actually need to know to use Claude well — from your first message to running an entire AI workforce.

WRITTEN BY JONATAN BLUM

orbitagents.xyz
FREE CREDITS ON SIGNUP

HOW TO READ THIS

Every section builds on the last, simple to advanced. Skim what you know, slow down where it's new. If a section doesn't apply, skip it — nothing later depends on it.

IN THIS GUIDE

- 01 Meet Claude
- 02 The four essentials
- 03 Claude that does the work
- 04 The honest limits
- 05 Running the whole thing

PART ONE

What is Claude?

Claude is a highly advanced, conversational artificial intelligence assistant and large language model (LLM) developed by Anthropic. It is built to help users with tasks like writing, coding, research, and data analysis through natural, human-like dialogue.

1.1 Claude is like one mind living in many rooms

One intelligence you reach in several places, each built for a different kind of work.

**Chat**

The front door — conversation, thinking, writing.

**Projects**

Chats grouped with shared instructions & memory.

**Cowork**

Operates on your files, folders and apps.

**Claude Code**

The same power, built for developers.

**Dispatch**

Control your desktop Cowork from your phone.

**Routines**

Scheduled cloud jobs — laptop closed.

THE ONE LINE TO REMEMBER

Chat is for thinking *with* Claude. The other rooms are for Claude *doing things* for you.

**New to Claude? Don't learn all six at once.**

Start in the plain chat window and stay there until it feels natural — it's the same Claude behind every other room. Reach for a Project only when you notice yourself re-explaining the same background, and for Cowork only when you actually want Claude to open and change your files. The rooms are there the moment a real need shows up; there's no prize for using them early.

1.2 Which room to open, for which job

Just chatting or thinking and strategy	→ The normal chat window
Something you return to often	→ A Project
Real work on your files or Browser	→ Cowork
You write software or connect to APIs	→ Claude Code
Away from your desk	→ Dispatch
Want it to run on a timer	→ Routines (code) - Cowork (Schedule)

1.3 There isn't one Claude – there are four models

Claude comes in different “sizes.” Picking the right one saves money and quality.



Fable 5

THE GENIUS

Hard strategy where quality is the whole point.



Opus 4.8

THE DEEP THINKER

Serious writing, research, complex work.



Sonnet 4.6

EVERYDAY DRIVER

Most of your work — fast, smart, cheaper.



Haiku 4.5

THE SPRINTER

Quick, high-volume, repetitive jobs.

THE MONEY RULE

Using the genius for everything is like taking a private jet to the corner shop. Run background work on Sonnet or Haiku; save Fable and Opus for the hard stuff. **This is the single biggest way to control cost.**

1.4 The one limit that matters: context

Claude reads your whole conversation plus everything attached — up to a large but finite window.



Long chats get fuzzy

If it starts feeling forgetful — it is. Middle details slip.

New chats start blank

A fresh window remembers nothing. Projects & memory help.

THE TRICK THAT ALWAYS WORKS

When a long chat gets fuzzy, ask Claude to *“summarise everything we've decided so far.”* Copy it, open a fresh chat, paste it in — a clean, sharp Claude with all the context and none of the clutter.

1.5 Plans, briefly

Free

The basics, limited daily use. Fine for tasting.

Pro

~\$20/mo

The real starting point: Projects, Skills, Cowork, Routines, Code.

Max

~\$100+

Heavy-usage headroom, earliest access to new features.

Team

Shared workspaces & admin controls. Some features need enabling.

All plans meter usage over rolling windows. Heavy agentic work burns far more than chat. Hit a limit? It resets on a schedule, smaller models use less, and trimming what you attach helps.

PART TWO

Four important things worth learning

Master these four and you're past 90% of Claude users.



2.1 1. Projects — stop re-explaining yourself

A container for one part of your life: chats + standing instructions + reference files + its own memory, shared automatically.

Instructions

Standing orders every chat follows.

Knowledge

Durable docs Claude always knows.

Memory

Its own memory, separate per project.

The magic move: put your best prompts into the project instructions — teach it once instead of every time.



2.2 2. Artifacts — ask for a *thing*, not just an answer

Claude builds a real, usable thing beside the chat: a doc, a working app, a calculator, a dashboard — editable, versioned, shareable. They can be AI-powered and save data across sessions.

Try it: *“Build me a simple meal planner I can use every week, and let it remember what I put in.”*

WHY THIS ONE MATTERS

The trick is to ask for the *deliverable*, not a description of it. “Explain how compound interest works” gets you a paragraph; “build me a compound-interest calculator with a slider” gets you a tool you can keep and share. Anything you'd otherwise rebuild by hand every week — a tracker, a planner, a quiz — is a candidate. And because artifacts version themselves, you can say “make it blue,” “add a column,” or “undo that” without ever starting over.



2.3 3. Skills — teach Claude once, it keeps it forever

Reusable know-how Claude loads when relevant — how to build your deck, your brand voice, your team's process. Some ship built in; you can create your own or install shared ones.



Watch out: pasting a skill's text into a chat works — but only for that chat. To keep it, install it properly in Settings.



2.4 4. Connectors & MCP — let Claude reach your apps

A connector lets Claude read from and act through your other tools — Orbitagents, Gmail, Calendar, Drive, Notion, Slack, GitHub, your CRM. The underlying tech is **MCP**, the “USB standard” of AI.

Connect before you need it

Agents can't inherit tools you never linked.

Scope deliberately

More tools ≠ better. Each surface can differ.

Treat a URL like a password

A personal endpoint lets anyone act as you.

That's the four things.

Projects, artifacts, skills and connectors.

Learning these set you apart from the res of the Claude users. You'll start excelling.

Moving on to part 3...

PART THREE

Understanding which Claude to do the work

3.1 Cowork vs Claude Code — same engine, two homes



Cowork

BUILT FOR

Knowledge work — docs, decks, sheets, ops.

OPERATES ON

Folders, apps, connectors — falls back to your mouse & keyboard.

TYPICAL ASK

"Summarise this folder of invoices into a sheet."



Claude Code

BUILT FOR

Software — repos, terminals, tests, PRs.

OPERATES ON

Git repositories, the command line, your toolchain.

TYPICAL ASK

"Fix the failing tests and open a PR."

Rule of thumb: non-developers want Cowork; developers want both. The folder you grant access to *becomes the agent's world* — give each recurring workflow its own.

3.2 Dispatch — your desktop, remote-controlled

Pair your phone to your desktop Cowork session once. Fire off a task from anywhere and your computer does it with your local files and logins, then pings you.

Asynchronous

Works while you're away and reports back.

Local execution

The phone is a remote control, not a second brain.

Context persists

Start on phone, continue at the desk.



The one catch: your computer must be on, awake, with Claude open. Laptop closes → Dispatch stops. For laptop-closed automation, use Routines.

3.3 Scheduled Tasks vs Routines – automation, two ways

	Local scheduled task	Cloud Routine
Runs on	Your machine (must be on)	Anthropic's cloud (laptop can be closed)
Access to	Full local world: files, MCP, skills, plugins	Only what you granted at creation
Memory between runs	None — fresh each run	None — fresh, isolated, zero context
Triggers	Schedule	Schedule (min 1 hr) · API · GitHub

1

Connectors are chosen per routine
Not inherited from your chats. The #1 failure is “why can't it see my calendar?” — it was never granted it.

2

Every run starts with amnesia
Any state that matters must live outside the routine — a file, a database, or a memory layer it reads each run.

3.4 Claude in the browser (and Excel, and PowerPoint)

When the work lives inside an app, Claude can drive the app.



Claude in Chrome
Navigates, reads, fills forms. The universal fallback.



Claude in Excel
Formulas, cleanup, models — in the sheet.



Claude in PowerPoint
Outlines into decks; edits in place.

Rule: connector first (faster, more reliable), browser second (the universal fallback).

PART FOUR

The honest bits: Where Claude misses

4.1 The line Claude won't cross

You'll want Claude to *automatically send* outreach — DMs on LinkedIn, X or Telegram. It refuses, even for one message. A deliberate safety line against messaging strangers; push on it and it's your account that gets restricted.

⊗ **Blocked**

Agents that autonomously *send* on social platforms.

✓ **Flows freely**

Agents that find, research, enrich, draft & *queue* — a human sends. Email to a consented list is normal.

Don't engineer around it. The sustainable design is “AI prepares everything, human fires” — better for anything sent in your name. The valuable part was never the click; it's the research before it.

4.2 The eight walls — where Claude structurally stops

None are fixed by better prompting. They're the shape of what one chatbot, alone, can do.

01 **Session amnesia**

Every new chat starts from zero.

02 **No cross-tool memory**

Claude doesn't know what Codex did.

03 **Amnesiac routines**

Scheduled runs forget every run.

04 **Output evaporates**

Work lands in logs you never reopen.

05 **No coordination**

Agents contradict each other, unnoticed.

06 **Decisions don't propagate**

Last week's decision gets ignored.

07 **You are the integration layer**

Copy-pasting context all day.

08 **Agents can't evolve**

A routine is fixed at creation.

Recognize three or more? Part Five is for you.

None of these are bugs, and none get better with a cleverer prompt. They're simply what happens when brilliant work has nowhere permanent to live — every session, every tool and every scheduled run starts over in its own little room. The fix isn't a smarter chatbot. It's giving your AI an *office*: one shared place to remember decisions, see what everyone did, and hold a single source of truth. That's what the last part is about.

PART FIVE

Running Claude operations optimally

5.1 The missing piece: Orbitagents.xyz

The eight issues mentioned on the last page share one root cause: Claude is a brilliant *worker*, but there's no *office* — no shared memory, no place to see what Claude, the agents or you yourself did, no single source of truth...



Orbitagents

Connect it to Claude once — paste one URL, ~2 minutes, no code. After that, every chat and agent checks in with the same shared brain: it reads what you've decided, sees what the others have done, and files its own work where you can find it.

One place to see everything done, remember what matters, and stay in control.

5.2 Every wall and every problem; solved

THE WALL	WITH ORBIT
Chats forget	every session starts by reading your memory + decisions
Tools don't talk	Claude, Codex, Cursor read and write one shared vault
Routines forget	they pull their state from Orbit at the start of each run
Output evaporates	a searchable vault + a morning briefing of the overnight run
Agents clash	they see each other's work; conflicts get flagged
Decisions ignored	decide once, and every agent honours it — forever
Agents can't evolve	change one instruction, every agent updates next run

5.3 Managing a fleet of scheduled agents

One routine is easy. Ten is a different problem. Each is fixed at creation, amnesiac and siloed — so a fleet quietly drifts: two agents do the same job, a third works from a decision you reversed last week, and nobody is watching the ones that failed at 3am.

WITHOUT A SHARED BRAIN

- Work duplicates — two agents chase the same lead.
- Silent failures — a run breaks and no one notices.
- Stale decisions — reversed choices keep running.
- Updates are manual — you edit each routine by hand.

WITH ORBITAGENTS

- Every run reads shared state first — no duplicate work.
- Live status on every agent — failures surface at once.
- Decisions are honoured everywhere, forever.
- Change one instruction — every agent updates next run.

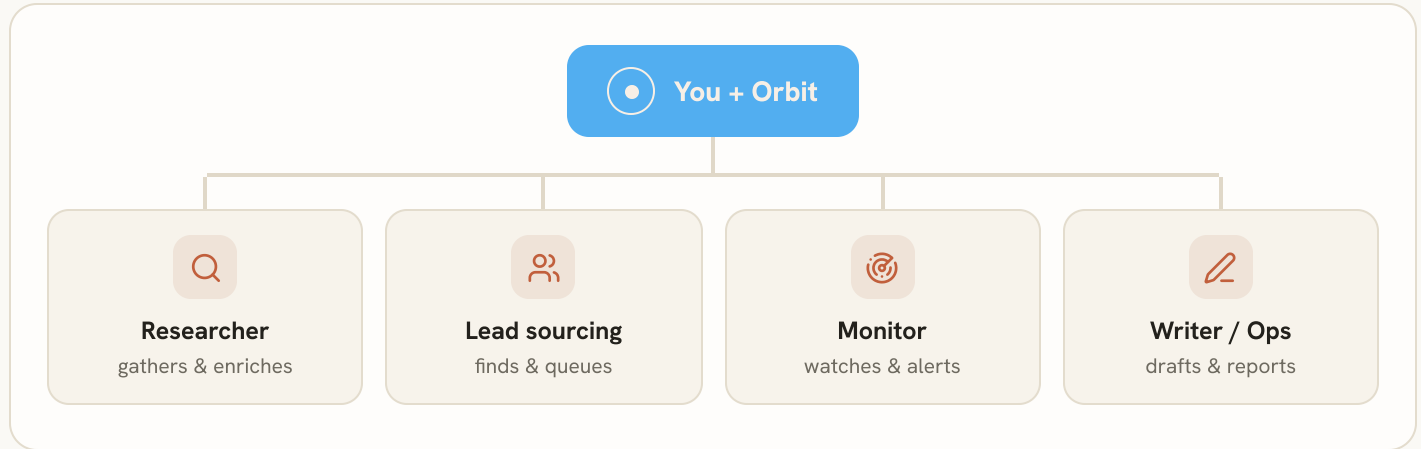
A MORNING OF SCHEDULED RUNS, AT A GLANCE

06:00	● Lead sourcing	42 new prospects enriched & queued
06:30	● Market monitor	3 competitor updates logged to vault
07:00	● Inbox triage	flagged 2 items for your decision
07:30	● Briefing writer	compiled overnight report — ready with coffee

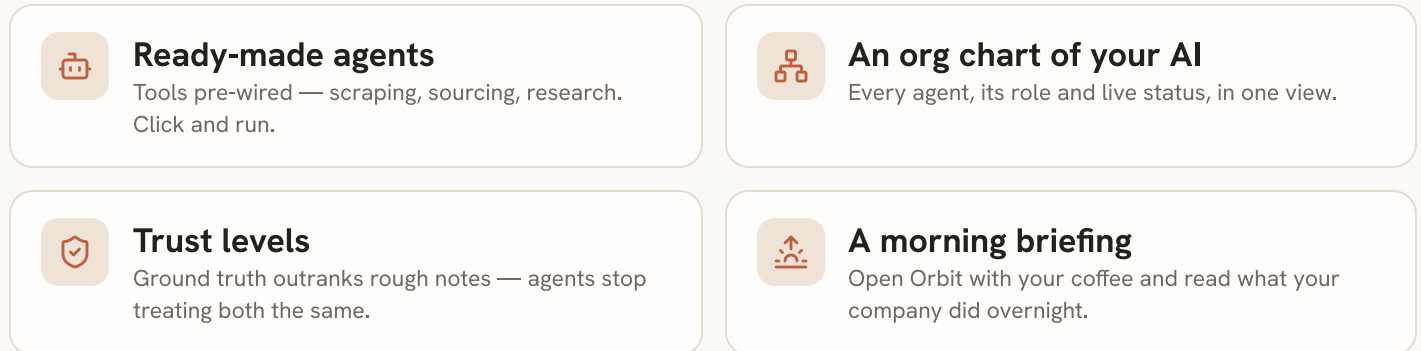
*The rule holds: state that matters lives outside the routine. Orbit is where it lives.
See everything from one place, read it like a brief. Manage it like a company.*

5.4 An entire workforce working for you

Once agents share one brain, you stop juggling chatbots and start running an organization. Everyone reports to the same place; you set direction and review the work.



WHAT ONLY THE OFFICE LAYER CAN GIVE YOU



It stops feeling like juggling chatbots. It starts feeling like running a company.

PART SIX

Getting more out of Claude

Small habits that quietly double the quality of what you get back.

6.1 Prompting that actually works

You don't need magic words. You need to tell Claude what you'd tell a sharp new colleague: who to be, what you want, and what “good” looks like.

**Give it a role**

“You're a tax accountant reviewing this” focuses the whole answer.

**Show the format you want**

Ask for a table, five bullets, or JSON — and that's what you'll get.

**Give one good example**

Paste a sample you like; Claude matches the pattern better than any rules.

**Say what NOT to do**

“No preamble, no bullet points” — constraints beat corrections.

**Let it think first**

“Reason it through before you answer” noticeably sharpens hard problems.

**Make it check itself**

“What did you miss? Poke holes in that” catches its own weak spots.

THE META-TIP

Stuck on how to ask? Ask Claude to write the prompt: *“Help me write a good prompt for X — ask me anything you need first.”* It's excellent at interviewing you into a better question.

6.2 Hidden habits that compound

The features people miss most aren't buried in a menu — they're small moves that change how the whole thing feels once they become reflex.



Edit, don't argue

Editing an earlier message re-runs the chat clean — far better than piling on corrections.



It reads pictures & PDFs

Drop in a screenshot, a chart, a whiteboard photo or a contract and ask about it.



Set a response style once

Tell it how you like answers — concise, no fluff, British spelling — and stop repeating yourself. Configure style in Orbitagents, see agents persist.



Fresh chat beats a long one

When it drifts, ask for a summary, open a new chat, paste it in. Sharp again instantly. More room for thinking



Attach, don't paste walls

Files stay tidy, reusable, and easier for Claude to reference than a giant pasted block.



Match the model to the task

Haiku for bulk and quick jobs; save Fable and Opus for the hard 10%. Switch mid-chat.

THE ONE THAT CHANGES EVERYTHING

Every tip on this page lives inside a single chat. The moment you want them to *persist* — a style Claude never forgets, decisions every agent honours, work you can find next week — you've outgrown the chat window. That's the whole argument for an office layer, and it's where this guide has been heading all along.



START HERE

YOU'VE GOT THE MAP

Build the AI empire that runs with you.

*The operating system for Claude, Codex, ChatGPT and more
— the office your AI work has been missing.*

Free credits on signup

~2-minute setup, no code

Works with the Claude you have

orbitagents.xyz →

WRITTEN BY AN OPERATOR WHO RUNS 29 AGENTS, A REAL COMPANY ON CLAUDE, EVERY DAY WITH
COMPLETE VISIBILITY AND EASE OF MIND!