Every Claude Cowork Concept Explained

The 20 concepts from the film, in the same 3 tiers and the same order, written for somebody who has only ever used a chat box. The permissions section is on this sheet rather than a separate one, because it is the part with real consequences.

20 CONCEPTS · 3 TIERS · IDEASREPAY.COM/ACADEMY/COWORK

BEFORE THE TWENTY

What Claude is

Claude is an artificial intelligence assistant built by a company called Anthropic. Most people meet Claude as a chat box: you type a question, Claude types an answer back, and that is the whole of it.

Three ways to use it

The website, which is that chat box in a browser tab. Claude Code, which is built for programmers and runs in a terminal window, meaning the plain typing screen programmers work in. And Cowork, which is this one.

What Cowork is

Cowork is the version of Claude that you can give access to your own computer. In the chat box, Claude cannot reach anything of yours. In Cowork you hand Claude one folder to work inside, and from that point Claude can open the documents in it, change them, write new ones and save the finished work back into the same folder.

Why it is called an agent

Because Claude can now reach your files and your programs, you stop asking Claude questions and start handing Claude jobs. The word for a program that works that way is an agent: it is given a goal instead of a question, and it is allowed to take steps on its own until that goal is met.

How you get it

Download the Claude desktop app from Anthropic, which is the version of Claude you install on your computer rather than open in a browser tab. Sign in on a paid plan, meaning Pro, Max, Team or Enterprise. Cowork is then there in the sidebar down the left.

1 Getting started

5 concepts

Where your work lives, and what Claude is allowed to know about it.

01

Work in a folder

The control that hands Claude one folder on your computer. You find it in the prompt bar, which is the box you type into, and it says work in a folder. You click it, you pick the folder, and from that moment Claude can open everything inside it. **Until you do that, Claude cannot see anything on your machine at all.** The folder you hand over has a name: it is the workspace folder.

IN PLAIN WORDS Attaching a file to a chat lets Claude read that one file. Giving Claude a folder lets Claude change what is inside it.

IN PRACTICE One client folder holds their contract, their last three invoices and your notes from the calls. You ask for this month's invoice at the same rates as last month. Claude opens the previous invoice to read those rates, opens the contract to check the terms, writes the new invoice and saves it into that folder named the same way as all the others. You did not open a single one of those files yourself.

WATCH OUT The first-day mistake is pointing Cowork at everything. Hand over the whole documents drive and Claude is reading six years of unrelated work to find the two files that mattered, which is slower and hands Claude far more of your material than the job needed. And if the files in a folder are the only copy you have, with no backup anywhere, copy them somewhere else first, because you are about to give something permission to change them.

02

Folder instructions (Claude.md)

A set of standing notes that belong to one folder, which Claude reads before starting any job in that folder. What they save you is repeating yourself: without them you type the same three sentences at the top of every session, telling Claude who the client is, what your file naming looks like, and what a finished piece looks like to you.

IN PLAIN WORDS You will also see this idea called Claude dot M D, a plain text file with that name sitting inside a folder. That is the same idea over in Claude Code, the version for programmers. In Cowork these notes are called folder instructions.

IN PRACTICE Five things go in them. Who the work is for. How files should be named. What the finished thing looks like. What Claude should never do. And anything about the material Claude would otherwise have to guess at. You are not the only one who writes on that card either: Claude can add to it while working, so something you explained once during a task is written down and is still there next week. A bookkeeper with eleven clients does not want to explain the house style eleven times, and this is where that style lives instead of in the bookkeeper's memory.

WHERE You edit them in the settings.

WATCH OUT Writing an essay. A page of instructions competes with the actual job for the room Claude has to think in. The version that works is short, specific, and about this folder rather than about you in general.

03

Global instructions

Standing notes that apply to every Cowork session you run, in any folder. Folder instructions only apply inside the one folder they live in, so the things that are true of every job you ever hand over need somewhere else to live, and this is it.

IN PLAIN WORDS The test for which of the two a note belongs in: ask whether it would still be true on a completely different job. Your job title is true everywhere. One client's file naming is not.

IN PRACTICE What belongs up here is about you rather than about the work. What you do for a living, so Claude stops explaining your own field back to you. How long you want things. Whether you want the reasoning or only the answer. One good line does more than people expect: somebody who writes that they are a bookkeeper stops being told what a profit and loss statement is in every answer, and gets the number they asked for instead.

WHERE Settings, under Cowork, next to the line that says global instructions.

WATCH OUT Anything tied to one piece of work does not belong here. A client's name, a deadline, the format one particular report goes out in: put those on the global sheet and they follow you into every unrelated job you open for the rest of the year. The other thing that goes wrong is writing a character brief instead of a working note. Three paragraphs about the tone you admire gives Claude less to go on than one sentence saying you write for builders and plumbers who want the price before the explanation.

04

Memory

What Claude keeps from your past conversations without you asking it to, so that Claude does not start from nothing every time. What gets saved is separate topics rather than a recording of the whole chat: one card for the client you keep mentioning, one for the software you use, one for how you like a report laid out.

IN PLAIN WORDS The difference between memory and instructions is who wrote them. You write instructions on purpose. Claude writes memory out of what happened, which means memory can be right about something you never intended to teach it.

IN PRACTICE Because Claude wrote it, you get to check it. You can open the list, read every topic Claude holds, and edit or delete any of them. Subjects like health, religion and politics are left out of memory by default, and switching those on is a choice you make rather than a setting that arrives already on. As of the film, the memory in the chat box and the memory in Cowork are now the same memory: something you told Claude in a chat is there when Cowork starts a task, and what Claude learns during a task comes back to the chat.

WATCH OUT Information goes stale. You tell Claude in March that a client wants everything on one page, the client changes their mind in August, and unless you edit that topic Claude is still working to March. Read the list now and then.

05

Context window

How much Claude can hold in front of it at one time during a single piece of work. It is measured in tokens. Everything Claude is working with counts against that size: your instructions, the memory Claude loaded, the files Claude opened, the results your tools handed back, and every word Claude has written so far in the task.

IN PLAIN WORDS A token is a chunk of text, usually a short word or a piece of a longer word, and it is the unit these systems count in instead of counting letters or words.

IN PRACTICE On the current models in Cowork the window holds a million tokens. On some of the older ones it holds two hundred thousand.

WATCH OUT When a long task fills that space, the oldest material stops being in front of Claude, and nothing announces that it has gone. That is what people are describing when they say Claude forgot the thing they said at the start. Claude did not forget the way a person forgets: the material went off the edge of what Claude can hold, and Claude carried on working from what was left. On a very long job, restate the things that must not be lost, or break the job into pieces that each start clean, rather than running one enormous session for six hours.

2 Capabilities

5 concepts

The things Claude can do once your work is in front of it.

06

Multimodal

The word for taking in more than text. Multimodal means Claude reads images, photographs, screenshots and PDF documents.

IN PLAIN WORDS A PDF is the fixed page file that scanners and invoices usually come out as.

IN PRACTICE The version you will use first is the screenshot: you photograph the error message on your screen, hand Claude the picture, and never have to describe what it said or spell out the code in it. The version that changes a working week is a pile of scanned invoices, where Claude reads the figures off pages that were never a spreadsheet and puts them into one. People find the third by accident, and it is the whiteboard: photograph the wall after a meeting, boxes and arrows and somebody's handwriting and all, and Claude gives you the plan written out as a document you can send. What all of it takes off you is transcription, which is a person being used as a cable between two systems that could not talk to each other.

WATCH OUT Handwriting and bad photographs. A clear scan is read accurately. A crumpled receipt photographed at an angle in poor light is where mistakes happen, so on figures that matter you check the total against the page rather than trusting what came out.

07

Web search

Claude going out to the live internet during your task, reading pages, and bringing back what it found. The reason it is needed is how Claude was made: Claude is trained on material up to a point in time and then the training stops, so anything that happened after that date is missing unless something goes out and fetches it.

IN PLAIN WORDS Whatever Claude brings back is attached to where it came from. You get the answer, and you get the pages Claude read, so a claim sitting in front of you can be opened and checked rather than taken on trust.

IN PRACTICE You ask what three competitors are charging. Claude searches, opens their pricing pages, pulls the figures, and writes the comparison into a document in your folder, and every number in that document has the page it came from sitting behind it.

WATCH OUT Having a source behind a number is not the same as the number being right. The link tells you where Claude got it, not that the page was correct, so on anything carrying money or risk the habit is to open the source rather than be satisfied that one exists. People also reach for search by reflex: if the answer is inside the folder on your own desk, searching the internet is the slower way round, and it brings back what the world says instead of what your own file says.

08

Extended thinking

Claude working a problem through in steps before writing the answer, and showing you those steps in a panel you can open.

IN PLAIN WORDS There are two separate controls sitting behind this, and telling them apart saves confusion. Thinking decides whether Claude works in the open in that panel. Effort decides how thorough Claude is on every reply, whether the panel is there or not. On the newest model, thinking cannot be switched off at all: it is how that model works rather than a feature you turn on.

IN PRACTICE The time is well spent on problems whose parts depend on each other. A plan where step four only makes sense if step two went a certain way is the shape that benefits. A request to reformat a list is not. The panel is the part people skip and the part worth opening: when an answer comes out wrong, the working shows you where it went wrong, and that is usually a wrong assumption near the start rather than a mistake at the end. The people who get the most out of raising these are the ones whose work has a wrong answer that costs something. A quote that has to be right. A schedule where one date moves everything after it.

WATCH OUT Turning everything up on a small job costs you time and allowance for an answer that was never going to be different, so raise it where a job has real consequences and leave it alone for the rest.

09

Artifacts

Something Claude makes that opens in its own window beside the conversation, instead of sitting inside it as a wall of text. Claude builds three kinds. A document, which is written work you will keep editing. A drawing, which comes out as a vector image. And a small working program, which is a thing you click and type into rather than read.

IN PLAIN WORDS A vector image is a picture that stays sharp at any size instead of going blocky when you enlarge it.

IN PRACTICE The third kind is the one that changes what you can ask Claude for. Ask for a budget tool and you get a budget tool with working arithmetic inside it, and you put your own numbers in rather than reading a description of one. What that replaces is the thing you were going to build in a spreadsheet over an evening: a pricing calculator, a staff rota, a small tracker, where the work was never the thinking but the fiddling with formulas. Inside Cowork these panels do one thing more, which is that they update while a task is running, so the thing in front of you changes as the work goes on rather than arriving at the end.

WATCH OUT An artifact is not a piece of software you have bought. It lives with the conversation. It is a good way to try an idea out and a good way to hand somebody a working version of one, and it is not where a business puts the system it depends on every day.

10

Bash tool

Claude running typed commands for itself in the middle of a job, rather than writing about them.

IN PLAIN WORDS Bash is the plain typed language a computer takes instructions in, one line at a time, underneath all the buttons and windows you normally click. You never have to learn any of it, and that is rather the point of it being here: you describe the outcome in plain English, and the commands are Claude's business, the way you order a meal without writing the recipe.

IN PRACTICE It is what Claude uses for the unglamorous middle of a job. Renaming four hundred files to a pattern. Pulling one column out of a large spreadsheet. Turning a folder of images into a single document. Somebody with three hundred photographs that all need renaming to a date and a location has an afternoon of clicking ahead of them, or one sentence.

WHERE Where those commands actually run depends on where the task is running. On your desktop Claude works on the folder you gave it. In a session running in the cloud Claude works inside a sealed container on Anthropic's servers, which is a separate machine holding none of your files unless they were sent there.

WATCH OUT These commands act on many files at once, which is where the approval setting earns its keep. A rename across four hundred files is four hundred changes, so the first time you hand over a job like that, run it on manual approval and read the plan before it goes.

3 Advanced

10 concepts

What Claude does when you are not sitting in front of it.

11

Projects

A walled-off workspace for one body of work. A project holds its own files, its own instructions and its own memory. The problem it solves is one job's settings turning up in another job: the way you like invoices laid out for a building firm arriving in the middle of the report you are writing for a dentist.

IN PLAIN WORDS Memory works the same way inside a project. Each project keeps a memory that belongs to it alone, so the more you use one project the sharper Claude gets inside its walls and the less it carries anywhere else.

IN PRACTICE You make one by picking a folder, naming the project, choosing where on your computer it is stored, and adding the instructions that belong to it. After that you open the project rather than going hunting for the folder. The person who feels the difference is anybody running more than about three streams of work at once: one client, one internal thing and one side project is the point where keeping it all in one place starts costing you time.

WATCH OUT Three limits worth knowing before you build anything on them. Projects live on your desktop and are stored on your own machine. There is no cloud copy of them. And on the team plans they cannot be shared with anybody else, so if you wanted to set a colleague up with the same working arrangement you have, a project is not how you hand it over today, and you would be writing the instructions out for them by hand.

12

Skills

A set of written instructions for one job you do again and again, saved so that Claude can pick it up and follow it. A skill sounds like programming and it is not: a skill is a folder with one text file in it, and that file is written in ordinary sentences, the way you would brief somebody joining your team on Monday.

IN PLAIN WORDS A skill takes the place of the checklist you wrote once and stopped opening. The difference between a skill and that document is that a skill gets carried out rather than remembered.

IN PRACTICE What a skill suits is the task with steps you always forget in the same order. Checking a deck before it goes out, formatting the monthly report, running the same five checks over a spreadsheet before anybody else sees it. Written out, a skill for checking a deck says what to look at, in what order, and what counts as a failure. Every slide has a heading. No slide has more than six lines. Every figure has a date beside it. The client's name is spelled the way they spell it.

WHERE You turn skills on under Customize in the sidebar, and the ones you write yourself stay private to your own account, unless you are on a team plan and choose to share them.

WATCH OUT The common mistake is building one enormous skill that does everything. Anthropic's own guidance says the opposite, which is separate skills for separate purposes, because a skill trying to cover four jobs gets picked for the wrong one.

13

Slash commands

The forward slash key followed by the name of a skill, and typing it runs that skill on the spot. Claude will pick a skill up on its own when your request matches it; a slash command is how you call one by name.

IN PLAIN WORDS One slash command comes built in already, and it is the one most people use first. Typing slash schedule inside a task sets that task to run again on its own.

IN PRACTICE The reason to name a skill rather than describe the job is certainty. Typing the name means you know which set of instructions is about to run, instead of hoping the words you chose matched the right one. It also saves you having to remember your own phrasing: without it you are recalling roughly how you asked last time and hoping it lands on the same instructions, and with it you press one key and read the list. Which makes what you call your skills matter more than it sounds. A skill called check is useless in a list of nine. A skill called check client deck before sending tells you what it does a year later, when you have forgotten writing it.

WATCH OUT The time not to reach for the slash is when you want Claude choosing. If you are not sure which of your skills fits, describing the job in your own words lets Claude match it, and that is the better move on anything you have not done before.

14

Plugins

A bundle. A plugin packs skills, connectors and sub-agents together into one thing that installs in a single action. What it gives you is somebody else's finished working setup, rather than their instructions for building one: you install the bundle and the whole arrangement is already wired together.

IN PLAIN WORDS Two of those words have their own entries further down this sheet. A connector is a link between Claude and another program you already use. A sub-agent is a second copy of Claude working alongside the first one.

IN PRACTICE The day this changes is the setup day. Somebody joins, and instead of a morning spent installing pieces and copying instructions into the right boxes, they install one bundle and start on the work.

WHERE You browse plugins in the same directory as skills and connectors, under Customize.

WATCH OUT A plugin is something somebody else wrote, so the same caution applies as installing anything else. It can carry connections to your own programs and instructions you have not read, so find out where it came from before it is running inside your work. And for one person doing one job on one machine, a plugin is more than the situation needs: a skill you wrote yourself does that job, and the bundle earns its place when the arrangement has to travel to somebody else.

15

Connectors

A link between Claude and another program you already use, so that Claude can read from that program and do things in it. Up to this point everything has happened inside one folder on one computer. **A connector is what takes Claude outside that boundary.** Each one is switched on by you, one at a time, and none of them is on until you switch it on.

IN PLAIN WORDS You will see these described as M C P connectors. It stands for the Model Context Protocol, an agreed way for a program to describe what it can do, so that any assistant can use it without somebody writing a special connection for that one pairing. You never touch the standard yourself. It is the reason the list of available programs keeps growing.

IN PRACTICE What it changes is the work that used to be assembly by hand. Take one Monday morning with a few connected: Claude reads the weekend's messages in the shared channel, opens the sales sheet, checks what is in your calendar for the week, and writes you the one page you were going to spend an hour building before your first meeting, without you carrying anything between the three yourself.

WATCH OUT The thing to think about before switching one on is what it opens up. Connecting your email gives Claude your email, so the sensible order is to connect the one program the job needs, rather than everything you own on the first afternoon.

16

Chrome extension (Claude in Chrome)

An add-on for the Chrome web browser that puts Claude in a panel beside the page, where Claude can read what is on that page and click through it for you. A connector works with a program that offers a connection, and most of the internet does not offer one. This is what covers the rest.

IN PLAIN WORDS There is a second way to get Claude onto a web page, and it helps to know which is which. Cowork now has a browser built into it, so a task can open a site without you installing anything at all. The Chrome add-on is the one you reach for when you want Claude working in the browser you are already sitting in.

IN PRACTICE It covers the sites that never built a connector, which is most of them. A supplier portal, a booking system, an internal tool with one login that nobody has updated since it was made. Inside those pages Claude fills forms, moves through steps and finishes tasks. Claude can take actions without asking permission for every single click, and each action is checked before it happens against what you asked for. The job this takes off somebody is the twenty-minute portal: log in every week, work through six screens, copy the same figures out.

WHERE If you looked for Claude in Chrome before and could not get it, there is a reason. As of the film it had just become available on every paid plan.

WATCH OUT What limits the add-on is where it runs. It does not work in other browsers built on the same underlying technology as Chrome, and it does not work on phones.

17

Computer use

Claude working your actual desktop. Claude looks at the screen, moves the pointer, clicks and types into the programs you have open, the way a person sitting at your machine would. A browser is one program on your screen, and the spreadsheet, the accounts package and the design tool sitting open next to it cannot be reached that way.

IN PLAIN WORDS Computer use is a research preview, which is Anthropic's own label for something released early and still being worked on. It runs on Mac and Windows, on the Pro and Max plans.

IN PRACTICE The reason it exists is the software that has no other way in. The old desktop program with no connector, no web version and no way to reach it except the buttons on the screen, which describes a great deal of what businesses actually run on. Worked through once: the stock system is a program from two thousand and nine that runs on your machine and nowhere else. Claude opens it, types the reference into the search box, reads the level off the screen, and writes that number into the spreadsheet in your folder. Practice management software, a warehouse system, the accounts package the business has run since before anybody thought about connections.

WATCH OUT Claude asks before touching each application, and some categories are shut off from the start, including investment and trading platforms and anything to do with cryptocurrency. It also differs from everything before it in one way that matters: when Claude runs code, it runs inside that sealed container. When Claude uses your computer, there is nothing between Claude and what is on your screen. Which makes the sensible guidance narrow. Computer use is for the task you cannot do any other way, on a machine where you are content to have Claude clicking, and it is not the tool to reach for when a connector would have done the job.

18

Scheduled tasks

A job you set once and it runs on its own after that, daily, weekly or monthly, at the time you picked.

IN PLAIN WORDS The part that surprises people is where these run. Scheduled tasks run in the cloud, on Anthropic's servers, so your computer does not have to be awake and the Claude app does not have to be open.

IN PRACTICE A scheduled task also gets everything a normal task gets, which means the programs you connected, the skills you turned on and the plugins you installed are all there when the task fires at six in the morning. The one worth setting up first is the report you build by hand every week: Claude reads the week's numbers, writes the summary in the shape you always write it, and it is finished before you are at your desk on Monday.

WHERE You set one with the slash schedule command, or from the scheduled list in the sidebar, and that same list is where you go to change one or switch it off.

WATCH OUT Where this goes wrong is the task that keeps running after the reason for it has gone. A weekly report on a project that finished in June is still arriving in September, so a look through that list every so often is the fix. Anthropic's own safety guidance carries a caution too, and it follows from the thing that makes them useful: a task running when nobody is watching is a task nobody is watching, so the ones you leave unattended should be the ones where a mistake costs you nothing.

19

Dispatch

Dispatch lets you send Claude a job from your phone and have it done on your desktop computer at home, using the files and the programs that are on that machine. Setting it up is a pairing between the two devices, and after that it behaves as one conversation: you send from the phone, you watch Claude work from either device, and you pick it up on the desktop later in the same thread.

IN PLAIN WORDS The difference between this and a scheduled task is the thing to carry out of both. Scheduled work runs on Anthropic's machines, so the lid can be shut. Dispatched work runs on yours, so it cannot.

IN PRACTICE The moment Dispatch is built for is the one on the train. You think of the thing that needs doing, your computer is at home with all your files on it, and the work starts now instead of at nine tomorrow when you are in front of it.

WATCH OUT One hard requirement: your computer has to be awake with the Claude desktop app open, because the work happens on that machine and nowhere else. It runs on the Pro and Max plans, it is called Dispatch rather than dispatch mode, and it holds one continuous thread, so there is no second conversation to start while the first one is going. And if your desktop runs Linux, which is a third operating system used alongside Windows and Mac, the jobs that need Claude to work your screen are not available over Dispatch.

20

Sub-agents

A copy of Claude working on one part of your job at the same time as other copies work on the others. Claude decides to do this on its own when a job is big enough to divide: you hand over one brief, and underneath it the work is being cut up and shared out without you arranging any of it.

IN PLAIN WORDS Each of those copies gets its own context window, the bounded working surface from concept five. That is the whole reason this is faster, because four separate surfaces hold four times as much as one.

IN PRACTICE The shape of job this suits is the one whose parts do not depend on each other. Researching six competitors is six jobs that never need to talk. Writing a document where each section follows from the one before it is one job, and cutting that up makes it worse. What comes back to you at the end is one finished piece of work, not four: the pieces are brought together before you ever see them.

WATCH OUT That speed has a cost, and it is a plain one. Running several copies at once uses your allowance several times over, because each copy is doing real work of its own.

Permissions: what Claude can do to your files

THE PART WITH CONSEQUENCES

Cowork has three settings that decide when Claude stops and asks you before it acts. You change them from the mode selector in the chat box, and you can change them in the middle of a task.

Setting 1

Manual approval

Claude pauses before acting, shows you what it is about to do, and waits for you to allow it or deny it. Nothing happens to your files until you press the button.

Setting 2

Automatic approval

Claude keeps working without stopping at every step, and instead of asking you, Claude checks each action itself for danger and blocks the ones it judges unsafe. This is where most people end up once they trust it.

Setting 3

Skip approvals

Nothing pauses and nothing checks, so that setting is the one to leave alone unless you trust every file, every program and every connection involved in that particular job.

ALWAYS ASKS Whichever of the three you are on, one action always asks. Claude has to get your explicit permission before permanently deleting a file, and you have to press allow before Claude can do it.

PROMPT INJECTION One of the dangers Claude is checking for on automatic approval has a name. Prompt injection is when instructions are hidden inside a document or a web page, written to be read by Claude rather than by you, telling Claude to do something you never asked for.

YOU CAN SEE IT You can see what Claude did. Cowork shows each step as Claude takes it, the files opened, the tools used and the choices made, and you can stop Claude and send a different instruction at any point.

WHAT TO WATCH FOR Anthropic's advice on watching Claude work is honest in the same way. You are not expected to check every single command Claude runs. You are expected to watch for the pattern that looks wrong, which is the thing a person is good at and a checklist is not.

THERE IS NO UNDO

There is no undo button. Anthropic's own advice is to keep backups, and to accept that some of what Claude can do is hard to reverse, like a message that has been sent or a purchase that has gone through. Keep backups of anything you cannot lose.

TWO HABITS FOR STARTING OUT

- Stay on manual approval until you have watched Claude do your kind of work a few times.
- Keep a copy of anything you cannot afford to lose.

If you open Cowork after this, do three things

And leave the rest alone until these are working.

1. Point it at one folder that holds a job you actually have.
2. Write three sentences of folder instructions into that folder.
3. Set it to manual approval, so you can watch what Claude does before Claude does it.

Every definition on this sheet is the one used in the film, and nothing has been added that the film does not say. Where the film gives a figure, the figure is here.