ANTHROPIC · 2026

From Beginner to Expert

The Complete Guide to Claude AI, Claude Cowork & Claude Code

Master the three most powerful AI productivity tools of 2026 — from your very first prompt to expert-level automation and agentic workflows.

Claude AI | Cowork | Code

maher@comverse.digital
comverse.digital
00216 98 795 559
4 Rue Dubai , Mourouj III
Ben Arous, 2074
COM VERSE - SUARL
1972507 M/A/M/000

Table of Contents

01 — Introduction: Why Claude in 2026?

The way we work has fundamentally changed. In 2026, AI is not a novelty — it's the operating system of the modern professional. The teams and individuals who learn to work with AI fluently will outperform those who don't, not by a small margin, but by an order of magnitude.

Anthropic built Claude with a single mission: to be the most genuinely helpful, honest, and safe AI assistant available. Unlike generic tools, Claude is designed to reason deeply, understand nuance, and adapt to your unique context.

This guide will take you from your very first interaction with Claude all the way to building complex, automated AI-powered workflows — step by step, with real-world examples for every level.

The Three Tools, Three Superpowers

Tool	Interface	Best For	Skill Level
Claude AI	Web, Mobile, Desktop	Writing, analysis, research, conversation	Everyone
Claude Cowork	Desktop App	File automation, workflow orchestration	Non-technical users
Claude Code	Terminal / CLI	Code generation, refactoring, agentic dev	Developers

Your Learning Journey

This guide is structured in four progressive levels. You can start anywhere, but we recommend following the path:

BEGINNER	INTERMEDIATE	ADVANCED	EXPERT
Weeks 1–2	Weeks 3–4	Month 2	Month 3+

PART 1: CLAUDE AI

Your intelligent assistant — available to everyone, everywhere

The Interface & Your First Steps

Claude AI is accessible at claude.ai via your web browser, or via the Claude mobile app (iOS and Android) and desktop app (Mac and Windows). You do not need a paid plan to start — the free tier gives you access to a highly capable model.

Navigating the Interface

When you open Claude, you'll see a clean chat window. Here's what matters:

- The message input bar at the bottom — where you type your prompts
- The attachment icon — lets you upload files (PDFs, images, spreadsheets, code)
- The web search toggle — enables Claude to fetch real-time information
- The Projects sidebar — your workspaces with persistent memory (see Advanced section)
- The New Chat button — starts a fresh conversation with no prior context

NOTE

Claude does not remember previous conversations by default. Each new chat starts from scratch — unless you use the Projects feature, which is covered in the Advanced section.

Your First Prompt — Making It Work Immediately

Most beginners write vague prompts and get vague results. The secret to great output is a well-structured prompt. Even as a beginner, follow this simple formula:

Role (optional)	Tell Claude what role to take: 'Act as a senior marketing consultant'
Task	What you want done: 'Write a LinkedIn post'
Context	The details that matter: 'about our new product launch'
Format	How you want it: 'in 3 short paragraphs, professional tone'
Constraint	What to avoid: 'no buzzwords, no emojis'

Mastering Prompts — With Real Examples

Prompt quality is the single most impactful skill you can develop. Here are the principles that separate beginner prompts from expert ones.

Principle 1: Be Specific

BEFORE VS AFTER — VAGUE VS SPECIFIC

You:

"Write me an email."

Claude:

I need more context! Who is it to? What's the purpose? What tone? Give me the specifics and I'll write something excellent.

BETTER PROMPT — SPECIFIC & STRUCTURED

You:

"Write a follow-up email to a potential investor, Sarah Chen, after a 30-minute intro call yesterday. We discussed our SaaS product for HR teams, our $2M ARR, and seeking $5M Series A. Tone: confident, warm, not pushy. Max 200 words."

Claude:

Subject: Great connecting yesterday, Sarah — following up on [YourCompany] Hi Sarah, Thank you for taking the time yesterday — it was genuinely energizing to hear your perspective on the HR tech landscape. [Continues with personalized, professional follow-up...]

Principle 2: Give Claude a Role

Telling Claude to 'act as' a specific expert dramatically improves output quality. Claude shifts its vocabulary, depth, and reasoning style based on the role you assign.

- 'Act as a senior UX researcher and review this landing page copy'
- 'You are a seasoned criminal defense attorney. Explain these contract clauses in plain English'
- 'Take the role of a skeptical investor and poke holes in my business plan'

TIP

You can stack roles: 'Act as an expert data scientist with 10 years in fintech and explain this to a non-technical CEO audience.'

Principle 3: Use Examples (Few-Shot Prompting)

Show Claude exactly what you want with examples. This is called 'few-shot prompting' and it's one of the most powerful techniques available.

FEW-SHOT EXAMPLE

You:

"Rewrite these product descriptions in our brand voice. Here are two examples of our voice: [example 1] / [example 2]. Now rewrite: [new description]"

Claude:

Using the voice from your examples — direct, confident, and human — here's the rewritten version: [output perfectly matching the provided style]

Advanced Features: Projects, Files & Web Search

Projects — Persistent Memory for Professionals

Projects is Claude's most powerful feature for professional use. A Project gives Claude persistent memory about you, your company, your preferences, and your ongoing work. It eliminates the need to re-explain context on every conversation.

How to Set Up a Project

1

Create a new Project

Click 'New Project' in the sidebar. Give it a meaningful name: 'Marketing Q2', 'Client: Acme Corp', 'Dev: Backend API'.

2

Write a Project System Prompt

In the Project settings, write a context block: who you are, your company, your tone preferences, key terminology, and what Claude should always/never do.

3

Upload Reference Documents

You can upload your brand guidelines, style guides, product specs, or any reference material. Claude will consult these automatically.

4

Start Chatting Inside the Project

All conversations within this Project share the same context. Claude remembers your preferences across sessions.

POWER

A well-configured Project saves 5–10 minutes per conversation and produces dramatically more consistent, accurate outputs. Set it up once, benefit forever.

File Analysis — Making Claude Read Your Documents

Claude can read and analyze uploaded files. This is transformative for professionals who deal with large volumes of information.

- PDFs: reports, contracts, research papers, financial statements
- Images: screenshots, diagrams, charts, photos
- Spreadsheets: CSV and Excel data (structured analysis, summaries, insights)
- Code files: review, debug, explain any code in any language

FILE ANALYSIS EXAMPLE

You:

"[Attached: Q3_Financial_Report.pdf] Identify the top 3 risks mentioned in this report, summarize the executive section in 5 bullet points, and flag any numbers that differ from Q2."

Claude:

Based on the Q3 Financial Report, I've identified: Top risks: 1) Supply chain exposure in Southeast Asia (mentioned 4x)... Executive summary: • Revenue grew 12% YoY to $84M • EBITDA margin compressed to 18%...

Web Search — Real-Time Intelligence

Enable the web search toggle to give Claude access to current information. This is essential for competitive research, news analysis, and any topic where freshness matters.

⚠ WHEN TO USE	Enable web search for: current events, competitor pricing, recent research, job market data, anything from the past 6 months. Disable it for: creative writing, private document analysis, purely logical tasks.

Claude AI — 8 Use Cases: Beginner to Expert

These use cases are ordered by complexity. Start with Level 1 if you're new to Claude. Each case includes exact prompts, step-by-step instructions, and tips to take it further.

USE CASE 1 ✍️ **Drafting Your First Professional Email**

Level: Beginner

Who is this for	Anyone new to AI tools — professionals, students, job seekers
The situation	You need to write a cold outreach email to a potential business partner but don't know where to start.

Step-by-step instructions

1. **Open claude.ai and start a new chat**
   No account is needed for a few free messages. For full access, create a free account at claude.ai.
2. **Describe the situation in plain language**
   Don't overthink it. Just tell Claude what you need: 'I want to email [Company X] about a potential collaboration. We make [product Y] and I think we could help their customers.'
3. **Ask for multiple versions**
   Add 'Give me 3 versions: formal, semi-formal, and friendly' — this lets you pick the tone that fits the relationship.
4. **Refine with follow-up instructions**
   Reply with feedback like 'The formal version is great but make the second paragraph shorter and add a specific call to action.'

TIP Always tell Claude the recipient's role/seniority. An email to a CEO reads differently from one to a developer. The more context, the better the output.

Result: A polished, personalized email in 3 different tones — ready to send in under 3 minutes.

USE CASE 2

Summarizing Long Documents

Level: Beginner

Who is this for	Analysts, managers, students, researchers — anyone dealing with information overload
The situation	You received a 60-page market research report. Your CEO wants a 1-page brief in 1 hour.

Step-by-step instructions

1. **Upload the document directly to Claude**
   Click the attachment icon in the chat and upload the PDF. Claude can handle documents up to several hundred pages.
2. **Use a structured summary prompt**
   'Summarize this document in the following structure: 1) Main thesis (2 sentences), 2) Top 5 findings with supporting data, 3) Risks and challenges, 4) Recommended actions.'
3. **Ask for an executive version**
   'Now rewrite this summary for a C-suite audience who has 2 minutes to read it. Maximum 200 words, no technical jargon.'
4. **Extract specific data on demand**
   'List every percentage figure mentioned related to market growth' or 'What does the report say about emerging markets?'

TIP For very long documents, ask Claude to start with a high-level overview, then drill into specific sections. This gives you a map before the details.

Result: A structured, executive-ready brief that would have taken 2 hours to write manually — done in 5 minutes.

USE CASE 3

Brainstorming & Ideation Sessions

Level: Intermediate

Who is this for	Product managers, marketers, entrepreneurs, creative teams
The situation	Your team is stuck in a creative rut. You need 50 content ideas for your newsletter in 30 minutes.

Step-by-step instructions

1 Set up the context clearly

'We run a newsletter for HR professionals about the future of work. Our audience is mid-to-senior HR leaders at companies of 200–2000 employees. They care about productivity, compliance, and culture.'

2 Ask for quantity first, quality second

'Give me 50 newsletter topic ideas. Don't filter — I want everything from obvious to provocative. Format as a numbered list.'

3 Cluster and prioritize

'Now group these 50 ideas into 5 categories based on theme and rank the top 3 in each category by potential engagement.'

4 Deep-dive into winners

'For idea #7, write a full content brief: working title, hook sentence, 3 key sections, and one contrarian angle to make it stand out.'

5 Create a content calendar

'Take the top 12 ideas and build a monthly content calendar for Q2 2026 with suggested publish dates and content types (article, case study, interview).'

TIP Use Claude as a thinking partner, not just a generator. Push back: 'These are too generic — give me 10 more that are more provocative or data-driven.'

Result: A full quarter of newsletter content planned, briefed, and calendared — in one focused session.

USE CASE 4

Advanced Data Analysis & Interpretation

Level: Intermediate

Who is this for	Finance teams, data analysts, consultants, business strategists
The situation	You have 3 months of sales data in a CSV and need to identify trends, anomalies, and actionable insights for Monday's board meeting.

Step-by-step instructions

1 Upload your CSV file

Drag and drop your data file into the chat. Claude can parse structured data and understand column headers, data types, and relationships.

2 Ask for a data overview first

'Describe this dataset: number of rows, key columns, data ranges, and any obvious anomalies or missing values you notice.'

3 Request trend analysis

'Identify the top 3 sales trends across this period. For each trend, cite the specific rows or data points that support it.'

4 Build the narrative for the board

'Write a 5-minute verbal presentation script that a VP of Sales would deliver to the board using these insights. Include transitions and anticipate 2 likely questions.'

5 Create an action plan

'Based on the underperforming segments, propose 3 specific actions we could take in Q2 with measurable success criteria for each.'

TIP Always ask Claude to cite specific data when making claims. Say 'Reference the exact figures from the data, not general observations' to get precise, credible output.

Result: A board-ready analysis with narrative, supporting data, and actionable recommendations — in one session.

USE CASE 5

Competitive Intelligence Research

Level: Advanced

Who is this for	Strategy teams, founders, product managers, investors
The situation	You need a comprehensive competitor analysis on 5 SaaS competitors before a product strategy meeting tomorrow.

Step-by-step instructions

1 Enable Web Search in Claude AI

Toggle on the web search icon. This allows Claude to access live data, current pricing pages, recent news, and public reviews.

2 Structure your research request

'Research these 5 companies: [A, B, C, D, E]. For each: current pricing, key features, recent product updates (last 6 months), G2/Trustpilot sentiment, and key weaknesses from user reviews.'

3 Build a comparison matrix

'Format all findings as a comparison matrix table with companies as columns and these criteria as rows: pricing, target customer, top 3 features, biggest weakness, recent news.'

4 Identify strategic gaps

'Based on this competitive landscape, identify 3 positioning gaps that our product could uniquely own. Support each with evidence from the research.'

5 Generate a strategic brief

'Write a 2-page competitive analysis brief suitable for sharing with the executive team. Include an executive summary, the comparison matrix, and 3 strategic recommendations.'

TIP

Cross-reference Claude's output by asking: 'Which of these findings are you most confident about and which should I verify independently?' This builds critical thinking into your workflow.

Result: A complete, board-ready competitive intelligence brief backed by real-time data — produced in 45 minutes instead of 2 days.

USE CASE 6

Legal Document Review & Simplification

Level: Advanced

Who is this for	Startup founders, freelancers, procurement teams, anyone signing contracts
The situation	You received a 15-page SaaS vendor contract with complex clauses. You need to understand your exposure before signing tomorrow.

Step-by-step instructions

1 Upload the contract PDF

Attach the document to your conversation. Note: Claude is not a lawyer — this is for comprehension assistance, not legal advice.

2 Ask for a risk overview

'Review this contract and flag the top 10 clauses that most favor the vendor over the customer. For each, explain the risk in plain English and rate it: High / Medium / Low.'

3 Deep-dive into critical clauses

'Explain the indemnification clause in Section 8 as if I'm a non-lawyer founder. What scenarios would trigger this and what's my maximum exposure?'

4 Compare to market standards

'Are these auto-renewal terms (Section 14) standard for SaaS contracts or unusually aggressive? What's a more balanced version?'

5 Generate negotiation talking points

'Write 5 specific negotiation requests I should make to the vendor's legal team, ordered by importance, with the business justification for each.'

TIP

Always confirm: 'Am I missing any high-risk clauses I should ask my lawyer about?' Claude will surface anything unusual. This doesn't replace a lawyer — it makes your lawyer's time 3x more efficient.

Result: Full comprehension of a complex contract, risk-ranked, with negotiation strategy — in 30 minutes versus 3 billable lawyer hours.

USE CASE 7

Building an AI-Powered Content System

Level: Expert

Who is this for	Content marketers, agencies, media teams, growth teams at scale
The situation	Your team publishes 20 pieces of content per week. Quality is inconsistent and production is bottlenecked by 2 senior writers.

Step-by-step instructions

1 Create a Master Brand Voice Document in a Project

In Claude Projects, upload your brand guidelines, 10 examples of best content, a negative examples document ('what we never write'), and a style guide. This becomes Claude's persistent reference.

2 Build a Content Brief Template

'Create a reusable content brief template for our blog. It should capture: target keyword, reader persona, key argument, 3 supporting points, CTA, and internal links to recommend.'

3 Create an Editorial Pipeline Prompt

'Given this brief [brief], produce: 1) A working title + 2 alternatives, 2) An SEO-optimized outline, 3) An introduction paragraph, 4) A conclusion with CTA. Use our brand voice reference.'

4 Build a Quality Checklist

'Act as our editorial director. Review this draft against our brand standards and score it 1–10 on: voice match, argument clarity, SEO optimization, and CTA strength. Suggest specific edits for anything below 8.'

5 Scale with a prompt library

Build a shared document with your 10 most-used prompts for your team. Each prompt should be tested, named, and annotated with when to use it.

TIP Invest 1 day building your Project and prompt library. The ROI is enormous: a team of 5 can produce content at the quality level of 2 senior writers, consistently, at 3x the speed.

Result: A scalable content production system that maintains brand quality at 3x output — with or without senior writers available.

USE CASE 8

Multi-Language Global Communication

Level: Expert

Who is this for	Global companies, international sales teams, multinational projects
The situation	Your company expanded to 4 new markets. All external communications — proposals, emails, marketing copy — need to be culturally adapted, not just translated.

Step-by-step instructions

1 Establish the cultural context

'You are localizing content for the Japanese market. Beyond translation, what are the key cultural considerations for B2B communication in Japan vs the US? Give me a cultural briefing.'

2 Set up market-specific Projects

Create a dedicated Claude Project for each market with: the language, cultural guidelines from step 1, local competitor names, market-specific terminology, and tone preferences.

3 Transcreate, don't just translate

'Transcreate this email for the German market — maintain the message and intent but adapt tone, directness, and examples to German B2B norms. Explain the 3 most significant changes you made and why.'

4 Create a multi-market review process

'Review this marketing headline in all 4 versions (English, French, Japanese, German) and flag any cultural risks, false friends, or tone mismatches.'

5 Build a cross-market terminology glossary

'Create a terminology glossary for our product in all 4 languages with approved translations, forbidden terms, and context notes for each market.'

TIP Always ask Claude to explain its cultural reasoning. 'Why did you change this phrase for the Japanese version?' builds your team's cultural intelligence over time.

Result: Professional, culturally-adapted communications for 4 markets — without hiring 4 translation agencies.

maher@comverse.digital
comverse.digital
00216 98 795 559
4 Rue Dubai , Mourouj III
Ben Arous, 2074
COM VERSE - SUARL
1972507 M/A/M/000

PART 2: CLAUDE COWORK

AI-powered desktop automation — for everyone, no coding needed

Installation & Setup

Claude Cowork is a desktop application (Mac and Windows) that brings AI-powered automation to your files, folders, and daily workflows. It's designed for professionals who want to automate repetitive tasks without writing a single line of code.

Getting Started in 4 Steps

1

Download the app

Visit claude.ai and look for the Cowork product. Download the installer for your operating system (Mac or Windows).

2

Sign in with your Anthropic account

Use the same account as claude.ai. Your Claude Pro or Teams subscription gives you access to Cowork.

3

Grant file system permissions

Cowork needs access to your folders to work. Grant it access to the directories you want it to manage — you remain in full control of what it can and cannot touch.

4

Run your first automation

Cowork will suggest starter automations based on your file structure. Try one to get a feel for how it works before building your own.

PRIVACY

Claude Cowork only accesses files and folders you explicitly authorize. It does not upload your files to external servers. Automations run locally on your machine.

Core Concepts

Automations

An automation is a task you describe in plain English that Cowork will execute automatically. Examples: 'Rename all PDFs in /Invoices with the format YYYY-MM_Vendor', 'Move any file older than 6 months to /Archive'.

Triggers

Triggers define when an automation runs: on a schedule (every Monday at 8am), when files arrive in a watched folder, or on demand with a single click.

Workflows

A workflow chains multiple automations together into a pipeline. Example: when a client sends files → rename → convert to PDF → move to client folder → generate a summary report.

Claude Cowork — 6 Use Cases: Beginner to Expert

USE CASE 1

Level: Beginner

Who is this for	Anyone with a chaotic Downloads folder, desktop, or shared drive
The situation	Your Downloads folder has 400 files accumulated over 6 months — receipts, screenshots, PDFs, zip files — with zero organization.

Step-by-step instructions

1. **Open Cowork and select 'New Automation'**
   Click '+ New Automation' and choose the 'Organize Files' template as a starting point.
2. **Point it to your target folder**
   Click 'Select Folder' and choose your Downloads folder. Cowork will scan the contents and show you what it finds.
3. **Describe your organization rules**
   In plain English: 'Sort by file type into subfolders: /PDFs, /Images, /Spreadsheets, /Other. Rename all PDFs with the prefix [YYYY-MM-DD_].'
4. **Preview before executing**
   Always click 'Preview' first. Cowork shows you exactly what it will do — every rename, every move — before making a single change.
5. **Run and verify**
   Click 'Run'. Check the activity log to confirm every action. A log is saved automatically for your records.

TIP Start with a non-critical folder for your first automation. Once you're comfortable with the preview system, apply it to more important directories.

Result: 400 files organized into a clean structure in under 2 minutes. A task that would have taken an afternoon of manual work.

USE CASE 2

Smart Email Attachment Handler

Level: Beginner

Who is this for	Professionals who receive dozens of email attachments daily
The situation	You receive 30+ email attachments per day that need to be saved, renamed, and filed into the right client folders — a daily 45-minute chore.

Step-by-step instructions

1. **Set up a watched 'Inbox' folder**

   Create a folder called /Inbox in your documents. Configure your email client to auto-save all attachments here.

2. **Create a Cowork automation on this folder**

   'Watch the /Inbox folder. When new files arrive: 1) Identify the client name from the filename or content, 2) Rename with [ClientName_YYYY-MM-DD_description], 3) Move to /Clients/[ClientName].'

3. **Handle edge cases**

   'If the client name is unclear, move the file to /Inbox/Review and flag it for my attention rather than guessing.'

4. **Set it as a trigger on file arrival**

   Set the trigger to 'On new file in folder' so it runs instantly every time something drops in — no scheduling needed.

TIP

Add a rule: 'If the file is an invoice, also create a line entry in my /Finance/Invoices_Log.csv with the date, client, and filename.' This gives you a running invoice log automatically.

Result: Your daily 45-minute attachment chore becomes a 0-minute background process that runs perfectly every time.

USE CASE 3

Automated Weekly Report Generation

Level: Intermediate

Who is this for	Team leads, department heads, project managers, analysts
The situation	Every Monday you spend 90 minutes pulling data from multiple sources, copying into a report template, and formatting it for the executive team.

Step-by-step instructions

1. **Identify and consolidate your data sources**

   Create a /WeeklyData folder where all input files land: export from your CRM, a CSV from your analytics tool, last week's action items doc.

2. **Build the automation pipeline**

   'Every Monday at 7:30am: Read the 3 files in /WeeklyData. Extract the KPIs: revenue, leads, conversion rate, top win, top risk. Generate a report in this format: [paste your template structure].'

3. **Add a formatting step**

   'Format the report as a professionally styled document. Use tables for numbers, highlight any metric that is more than 10% off from last week.'

4. **Auto-save to the right place**

   'Save the completed report to /Reports/Weekly with the filename format Weekly_Report_YYYY-MM-DD.docx. Archive last week's data files to /WeeklyData/Archive.'

TIP Give Cowork your last 3 manually written reports as examples. Tell it: 'Match this structure and tone exactly.' It will learn your format and replicate it consistently.

Result: Your 90-minute Monday report ritual becomes a file that's waiting in your Reports folder before you even open your laptop.

USE CASE 4

Multi-Format Document Conversion Pipeline

Level: Intermediate

Who is this for	Legal teams, finance departments, HR, agencies handling client documents
The situation	Your team receives documents in 6 different formats (DOCX, XLSX, PPTX, PNG, JPG, HTML) and everything needs to be converted to PDF for the archive system.

Step-by-step instructions

1 Create a conversion intake folder

Set up /ToConvert as a watched folder. Anyone on the team can drop files here.

2 Build the conversion automation

'Monitor /ToConvert. When files arrive: convert all DOCX and PPTX to PDF using standard formatting. Compress images into a single PDF if multiple arrive together. Leave already-PDF files untouched.'

3 Add quality checks

'After conversion: verify the output file is not empty (>0KB) and has the same number of pages as the original document. Flag any conversion failures to /ToConvert/Failed.'

4 Organize outputs by date and type

'Save converted PDFs to /Archive/[YYYY]/[MM] with original filename preserved. Keep a log entry in /Archive/conversion_log.csv.'

5 Set notifications

Configure Cowork to notify you via desktop notification when the batch is done and if any files failed.

TIP

Add a naming convention step: 'Before converting, rename the file to [YYYY-MM-DD_OriginalName] if it doesn't already start with a date.' This makes the archive searchable and chronological.

Result: A fully automated document archiving pipeline that runs silently in the background for your entire team.

USE CASE 5

Intelligent Contract Tracking System

Level: Advanced

Who is this for	Sales operations, legal teams, procurement, agency account managers
The situation	You manage 80+ active contracts. Renewal dates get missed, status is tracked in a spreadsheet that's always out of date, and no one has visibility.

Step-by-step instructions

1 Set up a structured contracts folder

Organize contracts into /Contracts/Active, /Contracts/Expiring, /Contracts/Expired. Each contract is a PDF named [Client_ContractType_StartDate].

2 Build the extraction automation

'Scan all PDFs in /Contracts/Active. For each contract, extract: client name, contract value, start date, end date, auto-renewal clause (yes/no), notice period required. Output to contracts_database.csv.'

3 Add the alert system

'Every Sunday evening: check contracts_database.csv for any contracts expiring within 30, 60, and 90 days. Create a priority report: /Reports/Contract_Alerts_[date].txt listing contracts by urgency.'

4 Build the status update workflow

'When a file is moved from /Active to /Expired or /Renewed, update contracts_database.csv automatically and add a timestamp.'

5 Create the dashboard input

'Generate a weekly summary table showing: total active contracts, total value, contracts expiring this month, and any high-value contracts (>$50K) expiring in 90 days.'

TIP

Add a clause: 'If any contract has an auto-renewal within 30 days AND no renewal/cancellation decision folder exists, create a /Decisions/[ClientName] folder with a decision checklist template.' This forces action before deadlines.

Result: A fully automated contract tracking system that eliminates missed renewals, with zero manual spreadsheet work.

USE CASE 6

Full Onboarding Automation Pipeline

Level: Expert

Who is this for	HR teams, operations managers, growing companies with frequent hiring
The situation	Each new hire triggers 40+ manual tasks: creating folders, generating documents, sending files, updating trackers. It takes HR 3 hours per new hire.

Step-by-step instructions

1 Build the trigger: New Hire Intake Form

Create a simple form (or watched CSV) where HR enters: new hire name, role, start date, department, manager. When saved to /HR/NewHires/, Cowork triggers immediately.

2 Create the folder structure automation

'When a new entry appears in /HR/NewHires/: Create /Employees/[Name] with subfolders: /Documents, /Onboarding, /Contracts, /Performance.'

3 Generate personalized documents

'Using the new hire data and our templates in /Templates/HR/, generate: Welcome Letter, IT Request Form, Benefits Enrollment Form, 90-Day Plan Template — all pre-filled with their name, role, start date, and manager.'

4 Populate the master HR tracker

'Add a new row to HR_Tracker.xlsx with: name, role, department, manager, start date, and status = Onboarding in Progress.'

5 Create the onboarding checklist

'Generate a checklist PDF for the manager at /Employees/[Name]/Onboarding/Manager_Checklist.pdf with 30 tasks pre-populated from our template, due dates calculated from the start date.'

6 Schedule follow-up triggers

'On day 30 and day 90 from start date: auto-generate a performance check-in form in /Employees/[Name]/Performance/ and update their status in HR_Tracker.xlsx.'

TIP

Build this once, then test with a sandbox hire first. Once confident, a single CSV entry triggers a fully automated, perfectly consistent onboarding experience every time — regardless of who is running HR that day.

☑ **Result:** What took 3 hours of manual HR work per hire now runs automatically in seconds, with zero missed steps and perfect consistency.

maher@comverse.digital
comverse.digital
00216 98 795 559
4 Rue Dubai , Mourouj III
Ben Arous, 2074
COM VERSE - SUARL
1972507 M/A/M/000

PART 3: CLAUDE CODE

The AI terminal copilot for developers — built for real codebases

Installation & First Commands

Claude Code is a command-line interface (CLI) tool that turns your terminal into an AI-powered development environment. Unlike browser-based tools, Claude Code works directly inside your project — reading files, running commands, and understanding your full codebase.

Prerequisites

- Node.js 18 or higher installed
- npm (comes with Node.js)
- An Anthropic account with API access
- A terminal (any OS: Mac, Linux, or Windows with WSL)

Installation

```
# Step 1: Install Claude Code globally
npm install -g @anthropic-ai/claude-code

# Step 2: Navigate to your project
cd /path/to/your/project

# Step 3: Launch Claude Code
claude

# You'll see the Claude Code prompt:
> Claude Code initialized. What would you like to build?
```

Command	What it does	When to use
claude	Launch Claude Code in current directory	Start every session
claude ask 'question'	Ask a single question without entering interactive mode	Quick lookups
claude review	Review current git staged changes	Before every commit
claude explain [file]	Get a plain-English explanation of any file	New codebase exploration
/clear	Clear conversation context	When starting a new task
/undo	Undo the last file change Claude made	Rollback mistakes
Ctrl+C	Cancel current operation	Stop any running command

Essential Commands — Quick Reference

Understanding Codebase Context

The single biggest advantage of Claude Code over browser-based AI tools is codebase context. Claude Code reads your entire project structure before responding — it doesn't just see the code you paste, it understands the architecture, dependencies, naming conventions, and patterns of your entire codebase.

How Context Works

- On launch, Claude Code indexes your project structure (filenames, imports, key definitions)
- It reads relevant files automatically when you ask questions about them
- It remembers the architectural patterns and conventions it observes
- It respects your .gitignore — it won't read files excluded from your repo

POWER TIP

Create a CLAUDE.md file in your project root. Write your project overview, architecture decisions, coding standards, and naming conventions here. Claude Code reads this file first on every session — it's your AI onboarding document.

MCP — Model Context Protocol Integrations

MCP allows Claude Code to connect with external tools and services — turning it from a coding assistant into a full development agent that can interact with your entire tech stack.

- GitHub/GitLab: Create PRs, review diffs, manage issues
- Databases: Query PostgreSQL, MySQL, MongoDB directly from your terminal
- Linear / Jira: Create and update tickets as you code
- Slack: Send updates and notifications from within your workflow
- AWS / GCP / Azure: Manage infrastructure without leaving your terminal

SETUP

Configure MCP servers in your claude_config.json or via the /mcp command inside Claude Code. Full documentation at docs.claude.com/en/docs/claude-code/overview

Claude Code — 6 Use Cases: Beginner to Expert

USE CASE 1 🔍 **Exploring & Understanding a New Codebase**

Level: Beginner

Who is this for	Developers joining a new team, open-source contributors, developers inheriting legacy code
The situation	You just joined a company and were handed a 50,000-line codebase with minimal documentation. You have a PR to review in 48 hours.

Step-by-step instructions

1. **Launch Claude Code in the project root**

   Run 'claude' from the root of the repository. Claude Code will index the entire project structure.

2. **Ask for a high-level architecture overview**

   'Give me an architectural overview of this codebase: main technologies, folder structure and what each folder is for, entry points, and the main data flow.'

3. **Drill into specific modules**

   'Explain the /auth directory in detail — how does authentication work, what pattern is used, and what are the dependencies?' Claude will read the actual files.

4. **Understand the data models**

   'List all the database models in this project, their relationships, and the most critical business logic attached to each.'

5. **Prepare for the code review**

   'I need to review PR #47. Explain what problem it's solving, what it changes, and flag anything that looks risky or inconsistent with the current codebase patterns.'

TIP

Ask Claude Code: 'What are the 3 things a new developer most commonly gets wrong in this codebase?' It will identify anti-patterns, gotchas, and non-obvious conventions from the code itself.

Result: Complete architectural understanding of a 50K-line codebase in 2 hours instead of 2 weeks of exploration.

USE CASE 2

Debugging Complex Issues

Level: Beginner

Who is this for	All developers — debugging is universal
The situation	A production bug appeared at 2am. Users are seeing a 500 error on checkout. The stack trace is 30 lines long and points to 5 different files.

Step-by-step instructions

1. **Paste the full error into Claude Code**

   Just paste the stack trace directly: 'Here is the error: [paste error]. What is causing this and where exactly in the code is the problem?'

2. **Let Claude trace the execution path**

   Claude Code reads the actual files mentioned in the stack trace and traces the execution path that led to the error.

3. **Ask for the root cause, not just the symptom**

   'Don't just tell me where it fails. Explain WHY it fails — what state or input is causing this and how did it get into that state?'

4. **Request a tested fix**

   'Provide the fix. Show me the before and after code diff. Also tell me: 1) What test I should write to prevent this from recurring, 2) Whether there are similar patterns elsewhere in the codebase that could fail the same way.'

5. **Ask for a post-mortem summary**

   'Write a 5-sentence incident summary I can share in Slack: what happened, root cause, the fix, and prevention steps.'

TIP When Claude gives a fix, always ask: 'Are there any edge cases in this fix I should test before deploying?' This catches 80% of 'fix that breaks something else' situations.

Result: Root cause identified, fix proposed, regression test written, and incident summary ready — in 15 minutes instead of a 2-hour debug marathon.

USE CASE 3

✨ Full Feature Implementation

Level: Intermediate

Who is this for	Mid-level to senior developers, full-stack engineers, indie hackers
The situation	Product just approved a new feature: a usage-based billing system with Stripe, usage tracking, and a customer dashboard. Estimated time: 5 days.

Step-by-step instructions

1 — Start with architecture alignment

'I need to implement usage-based billing with Stripe. Before writing any code, outline your implementation plan: what files you'll create/modify, the database schema changes needed, and the API endpoints required. Wait for my approval before proceeding.'

2 — Implement in logical layers

'Implement the database layer first: migration files for usage_events and billing_periods tables. Follow our existing migration pattern in /db/migrations.'

3 — Build the service layer

'Now implement the UsageBillingService class. It needs: trackEvent(), calculatePeriodUsage(), and syncWithStripe() methods. Use our existing service patterns in /services.'

4 — Wire up the API layer

'Add the billing endpoints to our Express router. POST /usage/track, GET /usage/summary/:userId, POST /billing/sync. Follow our existing controller pattern and add input validation.'

5 — Generate test coverage

'Write comprehensive tests for UsageBillingService. Cover: happy path, Stripe API failures, concurrent events, and billing period edge cases. Use our existing Jest setup.'

TIP Always start complex features with a plan-first prompt. 'Before writing code, outline your approach' prevents Claude from going deep in the wrong direction. Plan first, build second.

Result: Complete feature implementation with tests and proper architecture — in 1 day instead of 5, with consistent code quality.

USE CASE 4

Large-Scale Codebase Refactoring

Level: Advanced

Who is this for	Senior developers, tech leads, engineering managers with technical debt
The situation	Your Node.js codebase has grown from 5K to 80K lines in 2 years. Callback hell, no TypeScript, inconsistent error handling, and zero test coverage. You need to modernize without breaking production.

Step-by-step instructions

1. **Generate a technical debt audit**

   'Audit this entire codebase for technical debt. Categorize issues as: Critical (breaks things now), High (will break things soon), Medium (slows development), Low (code quality). Give me a prioritized list with file locations.'

2. **Plan the TypeScript migration**

   'Create a step-by-step TypeScript migration plan that allows us to migrate incrementally — one module at a time — without breaking the running application. Start with the lowest-risk modules.'

3. **Migrate module by module**

   'Migrate /services/userService.js to TypeScript. Add proper types, convert callbacks to async/await, add JSDoc for any types that are complex. Make zero behavior changes — only structural improvements.'

4. **Add test coverage as you go**

   'Before we migrate /services/paymentService.js, write comprehensive tests for its current behavior. These tests will be our safety net during migration.'

TIP Run 'claude' with the prompt 'What is the highest-risk area of this codebase if we start refactoring?' before starting. Let Claude identify the landmines before you step on them.

Result: Safe, systematic modernization of a legacy codebase — module by module, with a test safety net, without production incidents.

USE CASE 5

Security Audit & Hardening

Level: Advanced

Who is this for	Security-conscious developers, companies preparing for SOC2/ISO27001, pre-launch teams
The situation	Your startup is 3 weeks from launch and you want to do a security review before going live with real user data.

Step-by-step instructions

1 — Run a full security scan

'Perform a comprehensive security audit of this codebase. Check for: SQL injection vectors, XSS vulnerabilities, insecure direct object references, authentication bypasses, sensitive data exposure, and dependency vulnerabilities.'

2 — Analyze authentication and authorization

'Deep-dive the authentication system. Map every route to its authorization check. Flag any routes that are: unauthenticated by accident, missing role checks, or susceptible to privilege escalation.'

3 — Audit data handling

'Review how user data is handled: what's logged (are any PII fields in logs?), what's stored unencrypted, and whether we're compliant with basic GDPR principles (right to deletion, data minimization).'

4 — Fix issues by severity

'Fix all Critical security issues first. For each fix, explain: the vulnerability, the attack vector, your fix, and a test to verify the fix works.'

5 — Generate a security report

'Write a security audit report that I can share with our board and enterprise customers. Include: findings by severity, fixes applied, remaining accepted risks, and recommended ongoing security practices.'

TIP Claude Code cannot replace a professional penetration test, but it catches 70–80% of common vulnerabilities before they reach production. Use it as a first pass, then engage a professional for critical systems.

Result: A comprehensive security audit with fixes applied and documentation ready — in 1 day, before a single user accesses production.

USE CASE 6

Building an Agentic Development Workflow

Level: Expert

Who is this for	Senior engineers, platform teams, DevOps engineers, engineering leads at scale
The situation	Your team of 8 developers spends 20% of their time on boilerplate: writing standard CRUD endpoints, generating test files, creating database migrations, and updating documentation. That's 1.5 days per developer per week.

Step-by-step instructions

1. **Create a CLAUDE.md specification file**

   Document your entire codebase architecture, naming conventions, patterns, and standards in CLAUDE.md. This becomes Claude Code's standing instruction set for every generation task.

2. **Build a feature scaffolding command**

   'Given a resource name and its fields as input, generate: 1) Database migration, 2) Model with validations, 3) Service with CRUD methods, 4) Controller with all REST endpoints, 5) Router registration, 6) Full test suite. Use our existing patterns exactly.'

3. **Integrate with Linear via MCP**

   'When I say IMPLEMENT [ticket-id], fetch the Linear ticket, read the acceptance criteria, generate an implementation plan for my review, then — after approval — implement, test, and prepare the PR description.'

4. **Automate code review prep**

   'Before every PR: 1) Check all new code against our CLAUDE.md standards, 2) Verify test coverage >80% for new files, 3) Update CHANGELOG.md, 4) Generate a PR description with context, changes, and testing instructions.'

5.

TIP

The ROI of this investment is enormous. A team of 8 developers each saving 1.5 days/week = 12 developer-days per week recovered. That's the equivalent of hiring a 2.5 person team — using Claude Code instead.

Result: A fully automated development workflow where boilerplate is instant, code review is standardized, and tickets go from Linear to ready-to-merge PR with minimal manual intervention.

06 — Expert Power Workflows: Combining All Three

The true power of Anthropic's AI suite emerges when you use Claude AI, Claude Cowork, and Claude Code together in connected workflows. Here are three expert-level integrated workflows.

Power Workflow 1: The Full Product Launch Workflow

Scenario: Launching a new software feature from idea to market in one sprint.

Stage	Task	Tool
Discovery	Competitive analysis, user story drafting, technical spec writing	**Claude AI**
Development	Feature implementation, tests, code review	**Claude Code**
Documentation	Auto-generate API docs, README updates, CHANGELOG	**Claude Code**
File Management	Organize release assets, screenshots, marketing files	**Claude Cowork**
Go-to-Market	Write launch emails, blog post, social content	**Claude AI**
Post-Launch	Gather feedback, generate summary report, archive files	**Claude AI + Cowork**

Power Workflow 2: The Research-to-Report Pipeline

Scenario: Turning raw information into polished deliverables for clients.

- Claude AI (web search): Research topic, gather sources, identify key themes
- Claude AI (analysis): Synthesize findings, identify contradictions, form thesis
- Claude AI (writing): Draft the report with citations and structured argument
- Claude Cowork: Rename, convert to PDF, archive to client folder, update delivery tracker
- Claude AI (comms): Draft the delivery email to the client with report summary

Power Workflow 3: The Developer Sprint Accelerator

Scenario: A 2-week sprint with 3 developers shipping at 2x normal velocity.

- Monday AM — Claude AI: Sprint planning, ticket refinement, acceptance criteria generation
- Daily coding — Claude Code: Feature implementation, debugging, code review
- Daily docs — Claude Code: Auto-update technical documentation as code changes
- Cowork: Organize daily standup notes, weekly reports, client update files
- Friday PM — Claude AI: Write sprint retrospective, stakeholder update email, next sprint prep

07 — The Prompt Engineering Handbook

This section is your quick reference for writing prompts that consistently produce exceptional output. Master these patterns and you will be in the top 5% of Claude users.

The 6 Core Prompting Patterns

Pattern 1: Role + Task + Context + Format

The most reliable all-purpose structure:

TEMPLATE 'Act as [role]. Your task is to [task]. Context: [relevant details]. Format your response as [specific format]. Constraints: [what to include/exclude].'

Pattern 2: Chain of Thought

For complex reasoning, ask Claude to show its work:

TEMPLATE 'Think step by step. Before giving your final answer, reason through: [the problem]. Show your reasoning, then give your conclusion.'

Pattern 3: Critique + Improve

For iterative refinement of any content:

TEMPLATE 'Review this [content] and identify: 1) The 3 weakest parts and why, 2) What is missing, 3) What should be cut. Then rewrite the improved version.'

Pattern 4: Multiple Perspectives

For decisions, strategy, and analysis:

TEMPLATE 'Analyze this [decision/plan] from 3 perspectives: 1) As an optimistic advocate, 2) As a critical skeptic, 3) As a pragmatic implementer. Then give your synthesis.'

Pattern 5: Constraint-Driven Generation

For focused, high-quality creative and technical output:

TEMPLATE	'Write [content] with these strict constraints: [list constraints]. If you cannot satisfy all constraints simultaneously, tell me which ones you had to trade off and why.'

Pattern 6: The Socratic Method

For learning and developing your own thinking:

TEMPLATE	'I believe [my position/plan]. Rather than agreeing, steelman the opposing view. Then help me refine my position to account for the strongest counterarguments.'

Common Mistakes & How to Fix Them

Mistake	Why it fails	The Fix
'Write me a blog post'	No context, no audience, no goal	'Write a 800-word blog post for [specific audience] about [specific topic] with the goal of [specific outcome]. Tone: [tone].'
'Make this better'	'Better' is undefined	'Improve this for: clarity (reduce jargon), persuasiveness (stronger CTA), and brevity (cut by 30%).'
Accepting first draft	First output = first draft, not final	Ask: 'What could you improve in this response if you rewrote it?' Then ask it to rewrite.
No format request	Gets walls of text	Always end with: 'Format as [bullet list/table/numbered steps/sections with headers].'
Starting over each time	Loses context each conversation	Use Projects for ongoing work. Invest 30 min setting up your Project context.

08 — Quick Reference Cheat Sheets

Claude AI — Most Useful Prompt Starters

Goal	Prompt Starter
Summarize a document	'Summarize this in [format], focusing on [aspect], for [audience].'
Draft professional content	'Act as [expert]. Write a [content type] that [goal]. Tone: [tone]. Length: [length].'
Analyze data	'Analyze this data. Identify: trends, anomalies, and 3 actionable insights with supporting figures.'
Research with web search	'Search for [topic]. Give me [number] findings with sources, ordered by relevance.'
Strategic planning	'Help me build a [plan/strategy]. Start by asking me the 5 most important questions you need answered before we start.'
Review and improve	'Review this [content]. Rate each element 1-10 and rewrite anything below 7.'
Brainstorming	'Generate [number] ideas for [goal]. No filtering — quantity over quality. Then rank top 10.'

Claude Code — Command Cheat Sheet

What you want	What to type or say
Understand any file	claude explain [filename] or 'Explain what /src/auth.js does'
Fix a bug	Paste the error + 'Fix this. Show before/after diff and write a regression test.'
Add a feature	'Plan then implement [feature]. Follow existing patterns. Generate tests.'
Refactor code	'Refactor [file/module] for [goal: readability/performance/standards]. No behavior changes.'

Code review	'Review these changes for: security, performance, code quality, and consistency with the codebase.'
Generate documentation	'Document [file/function] with JSDoc comments and update the README section on [topic].'
Create migration	'Create a database migration to [change]. Include rollback. Follow our migration format.'

Claude Cowork — Automation Templates

Automation Type	Template Phrase
File organization	'When files arrive in [folder], sort them by [rule] into [subfolders] and rename as [format].'
Scheduled report	'Every [day/time], read [files], extract [data], and generate [report format] in [output folder].'
Conversion pipeline	'Convert all [file type] in [folder] to [output type] and save to [destination].'
Data extraction	'Scan all [file types] in [folder], extract [data fields], and output to [CSV/spreadsheet].'
Alert system	'Check [data source] daily. If [condition], create [alert file/notification] in [location].'
Archive workflow	'After [trigger], move processed files to /Archive/[YYYY]/[MM] and log the action in [logfile].'

09 — Resources & Next Steps

You now have everything you need to go from your first prompt to expert-level AI-powered workflows. Here's how to continue your journey.

Official Resources

Resource	URL	What you'll find
Claude AI	claude.ai	Web, mobile, desktop interface, Projects, file analysis
Claude Docs	docs.claude.com	Full API documentation, Claude Code guides, MCP setup
Claude Code Docs	docs.claude.com/en/docs/claude-code/overview	Complete Claude Code reference, tutorials, MCP integrations
Anthropic News	anthropic.com/news	Latest model updates, new features, research
Support Center	support.claude.com	Help articles, plan information, troubleshooting

Your 30-Day Action Plan

Week 1

Master Claude AI basics

Complete Use Cases 1–3. Set up your first Project with a system prompt. Practice the Role + Task + Context + Format prompt structure every day.

Week 2

Develop intermediate skills

Complete Use Cases 4–5. Try the Chain of Thought and Critique + Improve patterns. Upload your first document for analysis.

Week 3

Start with Cowork or Code

Developers: install Claude Code and run Use Case 1. Non-developers: install Cowork and run Use Case 1. Get one workflow running automatically.

Week 4

Build your first integrated workflow

Combine at least two tools in a workflow that solves a real problem in your work. Document your custom prompts and automations.

GOAL

By the end of Month 1, you should be saving at least 5 hours per week with Claude tools. By Month 3, that number should be 15+ hours — enough time to work on what truly matters.