

Complete Guide to OpenClaw

Build AI Agents for Revenue Teams in 30 Days

Last updated: February 2026 | Reading time: ~20 min | Interactive lessons: 12

Learn OpenClaw IN OpenClaw!

Download the complete interactive course and learn by doing real tasks with your agent.

[Download Complete Guide](#)

Unzip → Open folder in OpenClaw → Tell your agent: "Read START-HERE.md and start lesson 1"

Table of Contents

Core Concepts

What Is OpenClaw?

Requirements & Setup

The Mental Shift

The "Done" Framework

Workflows

Setup & First Agent

Building Your Bot's SOUL

Memory & Context

Skills System

Sub-Agents

Browser Automation

Reference

25+ Use Cases That Work

Limitations

Safety Checklist

What Is OpenClaw?

OpenClaw is your personal AI agent that runs 24/7. Unlike ChatGPT (which forgets between sessions), OpenClaw remembers everything, automates workflows, and works while you sleep.

The Key Difference: OpenClaw produces actual work, not just text. It can send emails, track prices, analyze portfolios, monitor customers, and alert you when action is needed.

Think of it as hiring an employee who:

- Never forgets context
- Works 24/7 without breaks
- Costs pennies per task
- Scales instantly

Who Is This For?

This guide is for **CS Directors, Revenue Ops, and Sales Leaders** who want AI leverage without engineering teams.

You're a good fit if:

- You're drowning in manual follow-ups
- You miss renewal windows because of volume
- You react to churn instead of preventing it
- You want AI help but don't code

Requirements & Setup

What You Need

- **Computer:** Mac, Linux, or Raspberry Pi
- **API Keys:** Anthropic (Claude) or OpenAI (GPT)
- **Messaging App:** Telegram, Slack, or WhatsApp
- **30 minutes:** To get everything running

Quick Install

```
# Clone the repo
git clone https://github.com/openclaw/openclaw.git
cd openclaw

# Install dependencies
npm install

# Run setup wizard
npm run setup
```

✓ **Pro Tip:** Start with Telegram — it's the easiest to set up and works great for testing.

The Mental Shift

The biggest mistake people make with OpenClaw is treating it like ChatGPT.

✗ Wrong Approach (Chatbot)

Ask a question → get answer →
copy-paste

Stay in the conversation until done

One task at a time

✓ Right Approach (Employee)

Describe outcome → step away → come back to
finished work

Delegate and move on

Batch 3-5 tasks in one session

⚠ If you keep treating it like a chat interface, you'll get 10% of the value.

The "Done" Framework

The quality of your output is directly proportional to the quality of your input.

Before you delegate anything, answer three questions:

1. What does "done" look like?

If you can't picture the finished state, neither can your agent.

2. What context does the agent need?

- Your naming conventions?
- Your folder structure preferences?
- The difference between work and personal data?

3. What constraints matter?

- "Don't delete any files, just organize them."
- "Only process files from the last 90 days."
- "Keep original filenames but add date prefixes."

Golden Rule: Always say "don't delete anything" unless you specifically want deletions.

Interactive Lessons

The following 12 lessons teach you OpenClaw by doing real tasks together with your agent.

Lesson 1

Setup & First Agent

Install OpenClaw, connect messaging, send your first command, and build a daily briefing agent.

Lesson 2

File Operations

Content-aware file organization. Sort 300+ files by reading contents, not filenames.

Lesson 3

Building Your Bot's SOUL

Understand SOUL.md, USER.md, TOOLS.md. Give your agent personality and preferences.

Lesson 4

Memory & Context

How OpenClaw remembers across sessions. Daily logs, MEMORY.md, and persistence strategies.

Lesson 5

Lesson 6

Skills System

Enable built-in skills, create custom skills, and build reusable instruction sets.

Understanding Models

When to use Claude vs GPT vs Minimax. Cost optimization and quality tradeoffs.

Lesson 7

Sub-Agents

Spawn multiple workers to tackle tasks in parallel. 50 minutes → 5 minutes.

Lesson 8

CS Automation

Build agents that prevent churn: at-risk check-ins, renewal reminders, escalation monitoring.

Lesson 9

Revenue Workflows

Pipeline review agents, deal alerts, competitive intel, lead enrichment automation.

Lesson 10

Browser Automation

Control your browser with Claude. Web scraping, form filling, data extraction.

Lesson 11

Personal Assistant

Build agents that run your life: flight tracker, trip planner, investment monitor.

Lesson 12

Production Deployment

Take agents from prototype to production. Monitoring, debugging, scaling.

25+ Use Cases That Work

Customer Success

- Weekly at-risk customer check-ins

- Renewal reminder automation (30/60/90 day alerts)
- Escalation monitoring from support tickets
- Customer health score tracking
- Quarterly business review preparation

Revenue Operations

- Daily pipeline review and anomaly detection
- Lead enrichment automation
- Deal stage progression alerts
- Competitive intel monitoring (X/Reddit)
- Win/loss analysis from call transcripts

Personal Productivity

- Flight price tracker for specific routes
- Trip planning and itinerary building
- Investment portfolio monitoring
- News alerts for topics you care about
- Daily briefing (calendar + priorities)

Financial & Admin

- Turn receipt photos into expense spreadsheets
- Extract obligations from contract PDFs
- Monthly budget tracking from bank statements
- Invoice summarization and payment tracking

Research & Analysis

- Synthesize customer interview transcripts
- Competitive feature comparison

- Market research from scattered sources
- Literature review from research papers

Limitations

OpenClaw is powerful, but it's not magic. Here's what doesn't work well:

What Doesn't Work Well

- **YouTube transcript fetching** — API access is unreliable
- **Complex multi-step browser workflows** — Pages load weird, elements don't respond
- **Real-time stock trading** — Latency makes it impractical
- **Tasks requiring external service connectors** — Limited integration options

Browser automation is hit or miss. It works great sometimes, gets stuck other times. If Claude gets stuck, tell it to try again or skip that part.



Safety Checklist

Before Your First Session





- Create a dedicated work folder (not Documents or Desktop)
- Back up any important files
- Decide what OpenClaw is NOT allowed to delete
- Set usage alerts to avoid cost surprises

Every Session

- Be explicit about constraints in your prompt
- Avoid putting sensitive files in workspace (credentials, financial docs)

-  If using browser access, stick to trusted sites
-  Review agent outputs before sending externally

If Something Seems Off

-  Stop the task immediately
-  Check what files were modified
-  Look for unexpected behavior
-  Restore from backup if needed

Golden Rule: Always say "don't delete anything" unless you specifically want deletions. OpenClaw only deletes when instructed, but be explicit.

TL;DR

- **OpenClaw is a worker, not a chatbot.** Delegate outcomes, step away, come back to finished work.
- **Write better prompts.** Define "done," give context, specify constraints.
- **Start with a dedicated folder.** Limit access while learning. Keep backups.
- **Use sub-agents for complex work.** OpenClaw parallelizes automatically.
- **Queue tasks and batch.** 3-5 tasks in one session saves usage.
- **Safety is your responsibility.** Backup files, avoid sensitive data, monitor behavior.

The tool is early. Start learning now.

Ready to Start?

Download the complete course and learn by building real agents with hands-on lessons.

[Download Complete Guide](#)

Built with OpenClaw by [Tim Pittman](#)

© 2026 OpenClaw Guide. All rights reserved.