

Claude Code for Non-Coders

INFORMATION

Claude Code is an AI agent that runs in your terminal. Unlike AI chats, it executes actions on your machine — creates files, reads documents, runs commands, and chains multi-step tasks. Start it with: `claude`

CLAUDE.md — Persistent Memory

Your instructions file that Claude remembers across all sessions.

Global: `~/.claude/CLAUDE.md`

```
# About Me

- My Obsidian vault: ~/Documents/obsidian-vault/

- My Anki database: ~/Library/.../Anki2/

- My automations: ~/automations/
```

Per-Project: `./CLAUDE.md`

```
# Project: IronHack Workshop

- Output files go in ./output/

- Use Portuguese for student materials
```

SUCCESS

Every time you repeat instructions to Claude, add them to CLAUDE.md instead. Build it gradually.

Context Management

The Golden Rule: One task, one session, one focus.

Your context window (~200K tokens) is like a desk — everything takes space. Overload it and things fall off. Connect sessions through **files**, not conversation history.

Instruction Compression

1. Front-load preferences in CLAUDE.md (write once, not every prompt)
2. Use reference files: "Follow ./brand-guidelines.md"
3. Create commands: /anki-cards [URL] replaces a paragraph
4. Be specific, not exhaustive — trust the configuration

Essential Commands

- /cost — Check token usage
- /clear — Clear conversation context
- /compact — Summarize to free tokens
- /help — Show all commands
- claude --version — Check version
- Exit + claude — Start fresh session

Skills & Commands

Custom Commands (simple)

Save a .md file in ~/.claude/commands/ — invoke with /command-name

Skills (complex workflows)

```
~/.claude/skills/my-skill/
```

■■■■ SKILL.md (instructions)

■■■■ scripts/ (executable code)

■■■■ references/ (documentation)

■■■■ assets/ (templates)

CAUTION

If you do something 3+ times, make a command. If it gets complex, upgrade to a skill. You can even ask Claude to create skills for itself!

Workshop Demo Prompts

- "Create meeting-notes.md with sections for Attendees, Agenda, Discussion, Action Items"
- "Fetch <https://arxiv.org/abs/2503.10622> and summarize the key findings"
- "Read ./paper.pdf, translate to Portuguese, save as paper-summary-pt.md"
- "Create 10 Anki flashcards from my notes about prompt engineering"
- "Create a skill that translates PDF papers to Portuguese"

Best Practices

1. Start simple — build complexity gradually
2. Build CLAUDE.md iteratively as you discover patterns
3. One task, one session — connect through files
4. Verify AI outputs — always fact-check
5. 3x rule — do it 3+ times? Make a command or skill
6. Check community first — awesome-claude-code on GitHub

INFORMATION

Ironhack Workshop | Automata Learning Lab | automatalearninglab.thinkific.com

