

Installing the tools

Reading time: ~12 min · Prereqs: Lessons 01–03 · 3 interactive demos inside

In Lesson 03 we looked at the four *doorways* into AI: chat, API, CLI, and code tools. This lesson is where you actually walk through one of those doors. We'll cover four tools, from the absolutely-nothing-to-install option to a full AI coding agent you run from your terminal. You don't need all of them — you just need to pick the one that fits where you are right now.

The short version

- **Just want to use AI?** → claude.ai. Nothing to install. Skip to Section 1.
- **Want private, offline AI with a friendly interface?** → LM Studio. Section 2.
- **Comfortable in a terminal and want lean, scriptable local AI?** → Ollama. Section 3.
- **Want AI to help you write and edit code?** → Claude Code. Section 4.

Not sure which one? Use the wizard below before reading on.

Tool Picker Wizard

Answer three questions and we'll suggest the right starting point — with the exact install commands if needed.

Q1 — What's your operating system?

— pick one —



Q2 — How much RAM does your computer have?

— pick one —



Q3 — What do you want to do?

— pick one —



Find my tool

Reset

The recommendations below are starting points — you can always try more than one.

1. Claude.ai — the front door (nothing to install)

The simplest possible start with AI is also, frankly, the best one. Claude.ai is Anthropic's web interface for Claude — you open a browser, sign up, and start typing. That's the entire setup.

1. Go to claude.ai in any browser.
2. Sign up with email or a Google account.
3. Start typing. That's it.

There's a free tier that's more than enough for learning. Paid plans (Pro and Max) give you heavier use limits and access to the most capable models. There are also desktop and mobile apps if you'd rather have something in your dock or on your home screen — same login, same experience.

One important thing to understand: when you use Claude.ai, *your text travels to Anthropic's servers*. For most people and most tasks, that's completely fine. But if you're working with genuinely private information — medical records, confidential business data, anything you wouldn't email to a stranger — that's when the local options below become relevant.

Start here, no matter what

Even if you plan to run AI locally eventually, spend your first week with Claude.ai. You'll learn what good AI output looks like, what kinds of prompts work, and what you actually want to do — all without wrestling with setup. That context makes the local tools much easier to appreciate later.

2. LM Studio — local AI with a friendly app

LM Studio is a free desktop app for Mac, Windows, and Linux that lets you download and run open AI models *completely offline*, with a clean, chat-style interface. No command line, no coding, nothing scary. It's the gentlest possible way to try running AI on your own hardware.

Why would you want this? Two reasons: **privacy** (nothing leaves your machine) and **cost** (once a model is downloaded, running it is free — no API bills). The trade-off is quality: local models are good and improving fast, but they don't yet match the best cloud models on hard tasks. And they need RAM.

The hardware reality check

Here's the honest version. A model's size is measured in parameters (the internal numbers that store what the model learned). More parameters = smarter but heavier. The quick guide:

- **8 GB RAM:** small models (around 3B parameters). Works, but slowly. Good enough for simple questions and drafting.
- **16 GB RAM:** mid-size models (7B–13B). Comfortable for most everyday tasks.
- **32 GB+ RAM:** large models (30B+). Noticeably better quality; smooth performance.

A dedicated GPU — especially on an Apple Silicon Mac (M-series chip) — makes local models much faster. The M-series architecture shares memory between CPU and GPU, which is unusually good for running LLMs. If you have a modern Mac with an M chip, local AI is genuinely practical. On a modest Windows or Linux machine with no dedicated GPU, cloud AI is the better experience.

Hardware / Model-Size Matcher

Drag the slider to your RAM amount and see what you can comfortably run.

RAM:  16 GB

Small models (3B)

Mid models (7B–13B)

Large models (30B+)

8 GB RAM minimum
Slow but usable
~4–8 tokens/sec

16 GB RAM comfortable
Good for most tasks
~8–20 tokens/sec

32 GB+ RAM needed
Noticeably better quality
~20+ tokens/sec

At 16 GB RAM:

Largest comfortable model size: **Mid models (7B–13B params)**

Expected speed: **~8–20 tokens/sec**

16 GB is the sweet spot for most people. 7B–13B models like Llama 3.2 and Mistral 7B handle everyday writing, Q&A, summarisation, and simple code tasks well. Try `llama3.2` or `mistral`.

Tokens/sec estimates assume a modern CPU without a dedicated GPU. Apple Silicon and discrete GPUs can double or triple these numbers.

Installing LM Studio

1. Go to lmstudio.ai and download the version for your OS. It auto-detects.
2. Install it like any normal app and open it.
3. Open the **Discover** tab (magnifying-glass icon). This is a built-in browser of downloadable models. Each one shows its size and whether your hardware can handle it.
4. For a first model, pick something in the 3B–8B range. Click Download.
5. Once downloaded, open the **Chat** tab, load the model, and start chatting — fully offline, fully private.

Bonus: LM Studio as a local API server

LM Studio can run a local API server on port `1234` that mimics the standard OpenAI API format. That means apps which normally talk to a cloud AI can be pointed at your local one instead. You don't need this on day one, but it's there in Module 6 when you start writing code.

3. Ollama — local AI in the terminal

Ollama does the same job as LM Studio — download and run open models locally — but it lives entirely in the **terminal** (that black text window). If you've never used a terminal before, LM Studio is the friendlier path. But if you're comfortable typing commands, Ollama is faster to get going and much easier to automate.

Installing Ollama

Go to ollama.com/download and install for your OS. On Linux (or in WSL on Windows), you can use the one-line installer:

```
curl -fsSL https://ollama.com/install.sh | sh
```

Running your first model

Open your terminal (Terminal on macOS/Linux, PowerShell or Command Prompt on Windows, or a WSL window). Then type:

```
ollama run llama3.2
```

The first time, it downloads the model — a few gigabytes, so give it a minute. After that, it loads fast. You'll see a `>>>` prompt. Type your message and press Enter. Type `/bye` to exit.

The handy commands you'll actually use

```
ollama list          # see models you've downloaded
ollama pull mistral  # download a model without running it
ollama rm llama3.2   # delete a model to free up space
ollama serve         # run it as a background server
```

Like LM Studio, Ollama exposes a local API — by default on port `11434`, OpenAI-compatible. Module 6 shows how to use it.

LM Studio vs Ollama — the one-line summary

LM Studio is friendlier (clicks, no terminal). Ollama is leaner and better for scripting and servers. Most beginners try LM Studio first, then discover Ollama when they want to automate something.

4. Claude Code — an AI coding agent

Claude Code is a different kind of tool. It doesn't just *answer* — it *acts*. You run it inside any project folder on your computer, describe what you want in plain English, and it reads your files, writes or edits code, and runs commands on your behalf. It's what the previous lesson called a **code/agent tool**.

Because it can change real files, Claude Code is not the right first tool for a complete beginner with no coding background. But if you do write code — even a little — or if you're curious about where AI is heading, it's worth understanding.

What you need before installing

Claude Code requires a **paid Claude plan** (Pro, Max, Team, or Enterprise) or an **Anthropic Console account** with billing set up. The free Claude.ai tier doesn't include it. System requirements are light: macOS, Linux, or Windows, around 4 GB of RAM.

Two ways to install

Option A — Native installer (simplest, no extra software needed):

```
curl -fsSL https://claude.ai/install.sh | bash
```

Option B — Via npm (if you already have Node.js 18 or newer):

```
npm install -g @anthropic-ai/claude-code
```

On Windows, Option B works best inside WSL (Windows Subsystem for Linux).

After installing, navigate to any project folder in your terminal and type `claude`. It will ask you to authenticate in a browser window on first run. There are also extensions for VS Code and JetBrains if you'd rather use it inside your editor.

Use it on a test project first

Claude Code can edit real files and run real commands. Start it on a test folder or a project backed up with version control (git). Read what it proposes before approving. Think of it as a capable junior developer: you review the work before it goes live. More on agent safety in Module 9.

All four tools side by side

Tool	Runs where	Install?	Best for	Privacy
Claude.ai	Anthropic's cloud	No	Everyone, day one	Data leaves your machine
LM Studio	Your machine (GUI app)	App download	Private AI, no terminal	Fully offline
Ollama	Your machine (terminal)	Yes	Scripting, servers, developers	Fully offline
Claude Code	Your machine (terminal) + Anthropic's cloud	Yes	Coding, multi-step tasks	Code sent to Anthropic for processing

The install commands, all in one place

Use the cards below to copy the exact command for whichever tool you're installing.

Install commands

Click any Copy button to copy the command to your clipboard. Claude.ai requires no command — just a browser.

Claude.ai

No install needed. Open in any browser and sign up with email or Google.

LM Studio

Download the desktop app for Mac, Windows, or Linux. No command line needed.

```
https://claude.ai
```

```
https://lmstudio.ai
```

Ollama

One-liner installer for Linux (also works in WSL on Windows). Or download from ollama.com/download for Mac/Windows.

```
curl -fsSL https://ollama.com/install.sh | sh
```

Claude Code

Native installer (no extra dependencies). Requires a paid Claude plan or Anthropic Console account.

```
curl -fsSL https://claude.ai/install.sh | bash
```

Commands are copied as plain text. For Ollama and Claude Code, paste into your terminal and press Enter.

Where to go from here

Now that you have at least one tool installed, the most important next step isn't learning another tool — it's learning *how to talk to AI well*. That's what Lesson 05 is about: prompting. The quality of your results depends far more on how you phrase your request than on which tool you use. A well-crafted prompt to a free model often beats a vague prompt to the most powerful one.

Before you head to Lesson 05, try the homework for this lesson — it's designed to get you actually using whichever tool you just set up.

LAST-REVIEWED: 2026-06-04