

Glossary, cheat sheet, and next steps

Your permanent reference for the whole course — plus a clear path forward.

You've reached the final lesson. This page is designed to live in your bookmarks. It has the full glossary you can search, an interactive cheat sheet you can print, a 4-week action plan with checkboxes, and a curated list of where to go next.

How to use this page: Work through the content once. Then bookmark it. Come back when you need to look something up, print the cheat sheet, or check off your weekly progress.

The full glossary

Twenty-plus terms you've met across this course. Search by name or definition, or filter by category.

All

Concepts

Tools

Risks

Clear

AI	Software doing tasks that normally need human intelligence — understanding language, recognising images, answering questions, writing, planning, and predicting.	Concepts
Machine Learning (ML)	AI that learns patterns from data instead of hand-written rules. "Here are 10 million emails — figure out what spam looks like."	Concepts
Deep Learning	ML that uses multi-layer neural networks. Responsible for most of the progress in AI since ~2012.	Concepts

LLM

Large Language Model — a text-trained model like Claude, ChatGPT, or Gemini. Predicts the next token based on everything it has read.

[Concepts](#)**Generative AI**

AI that creates new content — text, images, audio, video, or code. LLMs are one type; image generators (Stable Diffusion, DALL·E) are another.

[Concepts](#)**Model**

The trained AI itself — the swappable "engine." Claude Opus, Llama 3, Gemini Flash are all different models.

[Concepts](#)**Parameters**

A model's internal knobs — "7B" means 7 billion parameters. More parameters generally means smarter but also heavier and slower.

[Concepts](#)**Prompt**

What you type to the AI. The whole craft of getting good results is learning to write better prompts.

[Concepts](#)**Token**

A chunk of text — roughly $\frac{3}{4}$ of a word. AI reads and charges by the token, not by words or characters.

[Concepts](#)**Context window**

How much text the model can consider at once. Past the limit, the model "forgets" the beginning of your conversation.

[Concepts](#)**Multimodal**

A model that handles more than text — also images, audio, video, or documents. Most modern frontier models are multimodal.

[Concepts](#)**Agent**

AI that takes multi-step actions toward a goal rather than just answering once. It can call tools, browse the web, or run code.

[Concepts](#)

RAG

Retrieval-Augmented Generation — feeding the AI relevant snippets of your own documents so it can answer from them instead of guessing.

Concepts**AGI**

Artificial General Intelligence — hypothetical human-level general AI that can do anything a human can. Does not exist yet.

Concepts**API**

Application Programming Interface — the way your programs send prompts to and get answers from an AI service over the internet.

Tools**API key**

Your secret password for using an API. Anyone who gets it can bill to your account. Store it in a `.env` file and never commit it to git.

Tools**CLI / terminal**

The text window where you type commands directly to your computer. Used to run Ollama, Python scripts, and Claude Code.

Tools**Local model**

A model running entirely on your own computer. No data leaves your machine — useful for privacy, offline work, or just experimenting for free.

Tools**Ollama**

A command-line tool for downloading and running local models. One command to download a model, one to run it.

Tools**LM Studio**

A desktop app (Mac/Windows/Linux) for running local AI models with a friendly GUI — no terminal required.

Tools**Claude Code**

Anthropic's AI coding tool that runs in your terminal and can read, write, and edit files in your project. Great for code generation and refactoring.

Tools**Hallucination**

Confident, false AI output — stated as fact, with the same fluent tone as correct answers. You cannot tell from tone alone. Always verify what matters.

Risks

Bias

AI inheriting unfair patterns from its training data — over-representing some groups, under-representing others, or reflecting historical discrimination.

Risks

Prompt injection

An attack where malicious text in input data tricks the AI into following hidden instructions — bypassing your intended instructions.

Risks

One-page cheat sheet

Everything you actually need to remember, compressed into six cards. Click any card to expand it. Hit **Expand all** then **⌘P** to print a one-pager.

To just use AI

Go to **claude.ai** and type. That's it.

Free tier is real — you can do useful work without spending anything. Create an account with a Google or email login and start a conversation.

Other options: **ChatGPT** (chatgpt.com), **Gemini** (gemini.google.com). They all work similarly from the chat interface.

To get good answers

The recipe: Role + Context + Task + Format.

"You are a friendly lawyer. I have a dispute with a landlord about a broken boiler [context]. Explain what rights I have under UK law [task]. Use bullet points, plain English, no jargon [format]."

Tips:

- Show an example output if you have one.
- If the first answer is wrong, say what was wrong and ask again — don't start over.
- Be specific. Vague question = vague answer.

To run AI privately on your computer

Easy route (GUI): Download [LM Studio](#), open it, search for a model (try `Llama-3.2-3B` for speed, `Mistral-7B` for quality), download, run.

Terminal route: Install [Ollama](#), then:

```
ollama pull llama3.2:3b
```

```
ollama run llama3.2:3b
```

Nothing leaves your computer. No API key. No costs.

To let AI write & edit code

Install Claude Code (requires Node.js 18+):

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

Then navigate to your project folder and run `claude`. It reads your files and can edit them directly.

Alternative: **Cursor** ([cursor.sh](#)) — a VS Code fork with AI built in; good for beginners who prefer a traditional editor.

To automate with code

1. Get an API key at [console.anthropic.com](#).
2. Store it in a `.env` file: `ANTHROPIC_API_KEY=sk-ant-...`
3. Install the SDK: `pip install anthropic python-dotenv`
4. Start with a cheap model (`claude-haiku-4-5`) and set `max_tokens` to cap costs.
5. Only upgrade to a larger model when you've confirmed the logic works.

Always (the three rules)

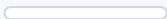
- 1. Verify what matters.** AI is a confident guesser, not a fact database. Check anything consequential against a real source.
- 2. Protect private data.** Don't paste passwords, patient records, or confidential company data into a cloud chat box.

3. Keep a human in the loop. For anything with real-world consequences — medical, legal, financial, safety-critical — a human makes the final call.

A suggested 4-week path

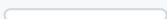
This is not a course. It's a practice plan. Each week adds one new capability. Check off tasks as you go — your progress is saved in your browser.

Week 1 — Chat only

 0%

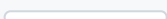
- ☐ Use Claude.ai every day this week for at least one real task (writing, research, summarising)
- ☐ Write a prompt using the full recipe (Role + Context + Task + Format)
- ☐ Take a mediocre answer and iterate on it with a follow-up until it's good
- ☐ Catch the AI making something up — verify with a real source and note the discrepancy
- ☐ Explain what an LLM is to someone who doesn't know — without jargon

Week 2 — Go local

 0%

- ☐ Install LM Studio or Ollama on your computer
- ☐ Download a small local model (3B–7B parameters)
- ☐ Ask the same question to both your local model and Claude.ai — compare quality
- ☐ Turn off your WiFi and confirm your local model still works

Week 3 — Touch the API

 0%

- ☐ Get an Anthropic API key at console.anthropic.com
- ☐ Store the key in a .env file (never in your code)
- ☐ Run the "Hello, Claude" Python example from Lesson 6
- ☐ Process a small batch of something with a loop (e.g., summarise 5 items from a list)
- ☐ Check your Anthropic usage console and understand what you spent

Week 4 — Build one tiny thing

 0%

- ☐ Pick a starter project from Lesson 8 (or write your own idea)
- ☐ Get a "hello world" working — even if it's ugly
- ☐ Make it do the one thing you actually wanted it to do
- ☐ Show it to one other person (even if they aren't technical)

Reset all progress

Where to go next

A curated selection of resources, grouped by what you want to do. All open in a new tab.

PROMPTING & BEST PRACTICES

[Anthropic Prompting Guide](#)



The official guide to prompt engineering with Claude — techniques, examples, and patterns.

[Prompt Engineering Guide](#)



Community-maintained comprehensive guide covering CoT, few-shot, RAG, and more.

[Learn Prompting](#)



Free, beginner-friendly course on prompt engineering — no code required.

API & SDKS

[Anthropic API Docs](#)



Complete reference for the Anthropic API — endpoints, parameters, SDKs, and examples.

[Claude Code Docs](#)



How to install and use Claude Code — the AI coding tool that lives in your terminal.

[Cursor](#)



VS Code fork with AI built in — great if you prefer a traditional editor over a terminal tool.

LOCAL AI

[Ollama](#)



The easiest way to download and run local models via the terminal. One command per model.

[LM Studio](#)



Desktop app for running local AI models — no terminal needed, friendly GUI.

[AnythingLLM](#)



Run a local RAG system — chat with your own documents using a local model. No data leaves your computer.

COMMUNITIES & LEARNING

[r/LocalLLaMA](#)



The best community for local AI news — model releases, benchmarks, and tips.

[Hugging Face](#)



The GitHub of AI models and datasets. Browse thousands of open models, run demos in your browser.

[Andrej Karpathy \(YouTube\)](#)



Deep but accessible explanations of how neural networks and LLMs actually work — from an OpenAI co-founder.

[fast.ai](#)



Free practical deep learning course — the best way to go from beginner to practitioner.

The whole secret: You don't need to understand how an engine works to drive a car. The same is true here. Start with Chat, build the verify-everything habit, and let curiosity pull you toward the next thing. That's it.