START HERE

What Is an LLM?

The plain-English guide to language models, local AI, and choosing your first model.



No maths. No jargon without an explanation. Just the mental models you need to start using AI with confidence.

The plain-English answer

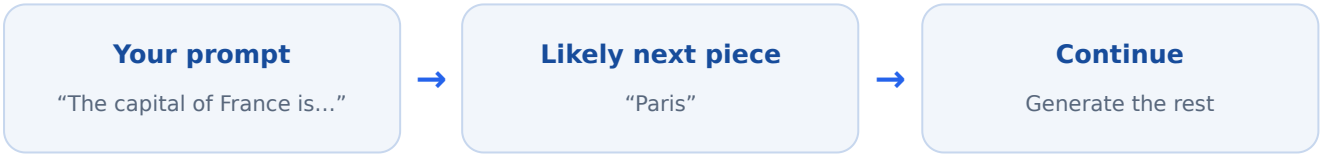
A **large language model (LLM)** is a computer program trained on a very large amount of language so it can generate a useful continuation to the text it receives.

You give it text—called a **prompt**—and it produces text back. That simple mechanism can look like conversation, explanation, translation, brainstorming, summarising, or computer programming.

> **An LLM is a remarkably capable language predictor—not a person, a search engine, or a database of guaranteed facts.**

A useful analogy: predictive text with a much bigger education

Your phone suggests the next word in a message. An LLM does a similar kind of prediction, but with vastly more training and context. It builds an answer one small piece at a time.



What it is good at

- Explaining and rewriting
- Finding patterns in provided text
- Drafting ideas and code
- Adapting tone and format

What it does not guarantee

- That every statement is true
- That it knows your private context
- That it understands like a human
- That its sources are real

THE HABIT WORTH LEARNING NOW

Treat an LLM as a fast, knowledgeable collaborator whose work still needs checking—especially for facts, code, health, money, and legal decisions.

What happens when you press send?

You do not need the mathematics to use an LLM well. A few simple ideas explain most of the experience.

1. **Your words become tokens.**
   A token is a small chunk of text—sometimes a word, part of a word, or punctuation. “Unbelievable” might be split into several tokens.

2. **The model reads the context.**
   This includes your message and, depending on the app, earlier messages or instructions supplied by the product.

3. **It predicts likely next tokens.**
   The model calculates probabilities based on patterns learned during training.

4. **Tokens stream back as an answer.**
   The app displays them as words and sentences. The process repeats until the model reaches a stopping point.

Context window

The amount of text the model can consider at once. A larger window can hold more conversation or documents, but it does not guarantee perfect recall or reasoning.

Parameters

The internal numerical settings learned during training. “7B” means roughly seven billion parameters. Size matters, but it is not the only measure of quality.

Training

The expensive learning process that creates the model. Everyday chatting is usually **inference**: using the trained model to generate a response.

Temperature

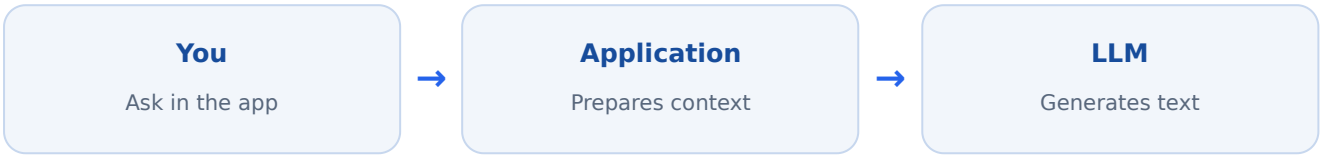
A setting that changes how adventurous token selection is. Lower is usually steadier; higher can be more varied. It is not a “truth” control.

WHY TWO ANSWERS CAN DIFFER

Model version, wording, context, settings, and selection randomness can all change the result. “Ask again” may produce a different answer—not necessarily a better one.

The model is not the app

A chat app is the visible product. The LLM is one engine inside it. The app may also add search, file handling, memory, safety systems, and other tools.



Two main places to run a model

Cloud AI	Local AI
Runs on a provider’s computers.	Runs on your own computer or server.
Usually quickest to start: open an account.	Needs a model download and suitable hardware.
Often offers the largest, newest models.	Usually uses smaller models that fit your machine.
Your prompts leave your device; read the privacy terms.	Prompts can stay on your machine when configured locally.
May have subscriptions, usage limits, or API charges.	No per-message provider fee; you supply power and hardware.
Normally needs internet access.	Can work offline after installation and download.

> **“Open weights” means you can obtain the model’s trained files. It does not automatically mean every right, training detail, or use case is unrestricted—check the licence.**

YOU CAN USE BOTH

Many people draft privately with a local model and use a cloud model for harder tasks. Agents can also switch providers without changing the whole workspace.

Three ways to run one locally

Ollama, LM Studio, and llama.cpp are model-running tools—not models themselves. They load a model and provide a way to chat with it or connect other software.

O

Ollama

A friendly command-line route with a local API. Popular with developers and a good first choice on Linux.

LM

LM Studio

A desktop app for discovering, downloading, and chatting with models. A gentle visual starting point.

C++

llama.cpp

A lightweight engine and building block used by many local-AI projects. Best when you want more control.

A gentle first path: LM Studio

1. Download LM Studio only from its official site and install it.
2. Use its model search to choose a small, beginner-friendly instruct/chat model that fits the app’s hardware guidance.
3. Download the model, open a chat, select it, and ask a simple question.
4. Explore its local-server option later, when you want another app or agent to use the model.

Prefer a terminal? Ollama looks like this

```
# After installing Ollama and choosing an available model:
ollama pull <model-name>
ollama run <model-name>
```

Model names and hardware recommendations change. Use the official catalogue rather than copying an old name from a guide.

HARDWARE REALITY CHECK

A model must fit in available memory, with room left for the system and context. Quantised models use less memory by storing numbers more compactly, often with some quality trade-off. A GPU can speed things up, but smaller models can run on a CPU.

Six services worth knowing

These are **popular starting points, not a permanent league table**. The model behind a product can change, so compare current features, price, privacy controls, and regional availability.

Three US-based providers

ChatGPT

OpenAI

General writing, learning, analysis, coding, and multimodal features.

chatgpt.com

Claude

Anthropic

Writing, analysis, coding, and working with supplied documents.

claude.ai

Gemini

Google

General assistance with connections to Google's ecosystem and multimodal features.

gemini.google.com

Three China-based providers

DeepSeek

DeepSeek

Chat, reasoning, and coding; its model family is also widely discussed for local use.

chat.deepseek.com

Qwen

Alibaba

General chat and multimodal tools; Qwen also offers many downloadable model variants.

chat.qwen.ai

Kimi

Moonshot AI

Writing, research, coding, and long-document assistance.

kimi.com

BEFORE YOU PASTE

Do not upload passwords, private keys, confidential client material, or sensitive personal data. Check the provider's current data-use and retention settings. A polished answer is not a privacy policy.

Start small. Be specific. Check.

You do not need magic prompt words. A good request usually supplies a goal, useful context, and the shape of the answer you want.

TRY THIS

“Explain what an LLM is to someone who understands email but has never used AI. Use one everyday analogy, keep it under 250 words, and finish with three safety tips.”

A simple prompt recipe

- ✓ **Goal:** what should it do?
- ✓ **Audience:** who is it for?
- ✓ **Context:** what should it use?
- ✓ **Format:** list, table, email, code?
- ✓ **Limits:** length, tone, exclusions?

Then improve the answer

- “Make that simpler.”
- “Show me a concrete example.”
- “What assumptions did you make?”
- “Point out anything I should verify.”
- “Give me two alternatives.”

Which option should you choose first?

If you want…	Start with…
The easiest first conversation	A reputable cloud chat service
A visual app on your own computer	LM Studio
A local model for other apps to call	Ollama or LM Studio’s server
A lightweight engine to build with	llama.cpp
Less powerful hardware	A smaller model—or cloud AI

THE 60-SECOND ACCURACY CHECK

Ask yourself: **What could be wrong? What is the source? Can I test it?** For code, run it safely. For facts, consult a primary source. For important decisions, ask a qualified human.

You now know the engine

An LLM turns context into a generated response. That is the engine. The next step is giving it a goal and tools so it can complete a task.

Five-word glossary

LLM
A large language model that generates text from context.

Prompt
The instructions or material you provide.

Token
A small chunk of text processed by a model.

Context window
The amount of material a model can consider at once.

Inference
Using a trained model to generate output.

Official places to explore

LM Studio
lmstudio.ai · Desktop local models

Ollama
ollama.com · Local runner and API

llama.cpp
Official source repository · Local inference engine

Hugging Face model hub
Discover model files and read their model cards and licences

YOUR FIRST SAFE EXPERIMENT

Ask two different services—or a local and a cloud model—to explain the same beginner topic. Compare clarity, examples, and claims. Verify one factual statement using an independent source.

Keep exploring the series

01 · LLM
The language engine

02 · Agent
A goal, tools, and a loop

03 · Harness
The agent's workspace