

Safety, privacy, cost, and limitations

Reading time: ~18 min · Prereqs: Lessons 01–08 · 4 interactive tools inside · The owner's manual you wish came with the chatbot.

The thing tutorials skip

Most "getting started with AI" guides show you the impressive parts. The creative writing that writes itself. The code that appears from thin air. The summary that takes seconds instead of hours. What they skip is the part of the owner's manual called *"Cautions and limitations."*

This lesson fills that gap. Seven topics. Each one grounded in something that actually goes wrong for real people. None of them should stop you from using AI — they should help you use it better and safer.

Use the four interactives

This page has four hands-on tools: a Privacy Risk Grader, a Cost Projector, a Hallucination Spotting quiz, and a Pre-flight Checklist. The checklist in particular is meant to be used every time you ship something AI-powered. Complete it before you move on.

1. Hallucinations: confident AI, wrong answer

You've heard the word before. Let's be precise about what it means.

A **hallucination** is when an AI model generates false information and presents it as fact — with complete confidence, fluent prose, and no disclaimer. It's not a bug in the traditional sense. It's a natural consequence of how LLMs work: they predict plausible text, and plausible text is not the same as verified truth.

The most dangerous hallucinations are specific ones. An AI claiming the speed of light is roughly 300,000 km/s is almost certainly right. An AI saying "According to a 2019 paper by Dr. Sarah Chen at MIT titled *Neural Pathway Optimisation in Transformer Architectures...*" is much more likely to be invented, down to the fake author, fake title, and fake institution.

The terrifying part is the tone. The AI sounds exactly the same whether it's right or fabricating. The confidence is baked in. You cannot listen your way to knowing if a claim is hallucinated — you have to check.

Making hallucinations less likely

Add these phrases to prompts where accuracy matters: "*If you're not sure, say so.*" and "*Please indicate your level of confidence.*" This doesn't eliminate hallucinations, but it prompts the model to hedge when it's guessing rather than presenting guesses as certainties. Also: for any specific claim (a statistic, a person's title, a publication), verify it against an independent source before you use it.

INTERACTIVE 1 OF 4 — SPOT THE HALLUCINATION

Each box below is a mock AI response. One sentence in each is the fabricated one. Click the sentence you think is the hallucination. You'll get instant feedback and an explanation.

RESPONSE 1 OF 4 — HISTORY QUESTION

You asked: "Tell me about the Battle of Waterloo."

The Battle of Waterloo was fought on 18 June 1815 near the town of Waterloo in present-day Belgium.

Napoleon's forces were defeated by the Duke of Wellington's coalition army, which included approximately 118,000 troops and was supported by a French naval blockade coordinated from Calais.

The battle effectively ended Napoleon's rule and led to his exile to the island of Saint Helena.

RESPONSE 2 OF 4 — SCIENCE QUESTION

You asked: "What causes rainbows?"

Rainbows form when sunlight enters water droplets in the atmosphere, bends (refracts) as it enters, reflects off the inside back surface, and refracts again as it exits.

The phenomenon was first mathematically described by Ibn al-Haytham in his 1021 work *Kitab al-Manazir*, which included precise measurements of the 42-degree arc angle now called the "al-Haytham constant."

Different wavelengths of light bend at slightly different angles, separating white light into the visible spectrum from red (outermost) to violet (innermost).

RESPONSE 3 OF 4 — TECH QUESTION

You asked: "What is Python used for?"

Python was created by Guido van Rossum and first released in 1991; it is consistently ranked as the most popular programming language in the TIOBE index, which measures language popularity based on a proprietary survey conducted exclusively with Fortune 500 companies.

It is widely used for data science, machine learning, web development, automation, and scientific computing.

Python's design philosophy emphasises code readability, and its syntax allows programmers to express concepts in fewer lines of code than languages like C++ or Java.

RESPONSE 4 OF 4 — HEALTH QUESTION

You asked: "Is coffee good or bad for you?"

Moderate coffee consumption (3–4 cups per day) has been associated in multiple large studies with reduced risk of type 2 diabetes, Parkinson's disease, and liver disease.

A landmark 2018 meta-analysis published in *The Lancet Nutritional Medicine* by Dr. Frank Hu's team at Harvard concluded that the optimal daily dose is precisely 3.7 cups and coined the term "the coffee sweet spot" which is now standard in dietary guidelines.

High caffeine intake can raise blood pressure and disrupt sleep, so people sensitive to caffeine or with certain heart conditions are generally advised to limit their intake.

Score: 0 / 4

Reset quiz

Notice how the fabricated details are always specific and plausible — that's what makes hallucinations dangerous. Vague statements are easy to question; specific false claims are easy to believe.

2. Privacy: what happens to your data

When you type into Claude.ai, ChatGPT, or any cloud AI service, your message travels over the internet to servers you don't control. Depending on the service's terms and your subscription type, that data may be:

- Stored and reviewed by company staff for safety and quality purposes
- Used to improve future model training (often opt-out on paid plans)
- Exposed in a security breach
- Subject to law enforcement requests

None of this means cloud AI is dangerous to use. It means you should think of it the same way you think of sending an email: suitable for most things, not suitable for secrets.

The practical rule: **never paste anything into a public AI chat that you would not put in a company-wide email or a public forum.** This includes passwords, API keys, medical records, financial account details, other people's private information (without their knowledge), and trade secrets.

When privacy is genuinely non-negotiable — you're a healthcare worker handling patient data, a lawyer handling client files, a developer working with production credentials — the answer is a **local model**. Ollama or LM Studio (Module 4) run entirely on your machine. Nothing leaves. The models are somewhat less capable than the top cloud models, but the privacy guarantee is absolute.

INTERACTIVE 2 OF 4 — PRIVACY RISK GRADER

Select the tool you're using and the type of data you're about to paste in. Get an instant verdict.

Tool

— Select tool —

Data type

— Select data type —

This grader encodes common-sense rules. When in doubt, prefer the local model or omit the sensitive detail entirely.

3. Bias: models learned from flawed humans

AI models are trained on text written by people. People hold biases — about gender, race, profession, nationality, and countless other dimensions. The model learns the patterns in that text, which means it absorbs the biases too.

This shows up in subtle ways: occupations assumed to belong to certain genders, certain nationalities associated with certain qualities, advice that reflects the assumptions of the dominant training corpus (which skews heavily English, Western, and internet-connected). More overtly, some models have shown different responses to identical questions framed around different demographic groups.

For casual use, this is worth being aware of but rarely a serious concern. For **decisions that affect people** — hiring, lending, insurance, medical triage — it matters enormously. Using AI to screen CVs at scale, for example, risks encoding and

amplifying whatever biases exist in the training data, applied to thousands of real people's careers.

The practical guidance: add human review and critical thinking wherever AI output touches individual people. Spot-check outputs for patterns. Don't automate at scale what you wouldn't trust on a single case. And know that many jurisdictions are actively legislating around AI in high-stakes decisions: the EU AI Act classifies AI for employment, education, and credit as "high-risk" with mandatory requirements.

Critical: audit before you scale

If you're building any workflow where AI output affects decisions about people, test it manually on 20–50 diverse examples before you automate. Look for patterns. Ask: "Would this output be the same if I changed the name from 'James' to 'Fatima'? From 'London' to 'Lagos'?" If the answer is no and it shouldn't be, you have a bias problem at scale.

4. Cost: tokens add up fast in a loop

Cloud AI APIs charge by the **token**. One token is roughly three-quarters of a word. The cost per token is tiny — fractions of a cent — which makes it easy to underestimate how quickly it accumulates when you're processing large documents, running many calls in a loop, or building an agent that makes requests autonomously.

A common mistake for new developers: writing a loop that processes a large dataset, forgetting to add a counter or cap, and waking up to a large bill. This isn't theoretical. It's a well-documented pattern in developer communities.

The fixes are simple but require doing them proactively:

- **Set a monthly spending limit** in your API console before writing a single line of code that calls an API. Anthropic Console, OpenAI dashboard, and Google AI Studio all have this.
- **Test with a tiny sample first.** Process 5 items. Check the cost. Then extrapolate. Don't run 10,000 items until you've confirmed the per-item cost is acceptable.

- **Use a cheaper model for development.** The smallest model in a family is often good enough for testing and costs a fraction of the flagship.
- **Local models are free per use.** Ollama and LM Studio run on your hardware. The cost is your electricity and the time to set them up once.

INTERACTIVE 3 OF 4 — COST PROJECTOR

Drag the sliders to see how API costs project over time. Then hit "Show runaway agent" for a shock.

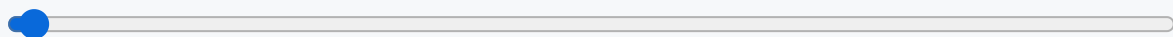
Cost per API call

0.10¢



Calls per day

100



DAILY COST

\$10.00

MONTHLY COST

\$300

YEARLY COST

\$3650

Show me a runaway agent ⚠

⚠ **This is real money.** Set a spending cap in your API console right now. Runaway loops at this scale have caused developers hundreds of dollars in unexpected charges overnight.

5. Agents need a leash

Earlier modules introduced AI agents: tools like Claude Code, AutoGPT, and others that go beyond answering questions to actually taking actions — editing files, running commands, sending requests, browsing the web.

The power here is real, and so is the risk. An agent doesn't have common sense about consequences. If you ask it to "clean up the project folder" and you've given it write access, it might delete things you didn't mean to delete. If you ask it to "send a follow-up email to everyone on the list" and you haven't reviewed the list, it might send 10,000 emails.

The failure mode isn't the agent going "rogue" in a sci-fi sense. It's the agent doing exactly what it understood you to mean — which wasn't quite what you meant. Misunderstood instructions at agent speed cause real damage.

The rules are few but firm:

- **Start on test environments or version-controlled code.** If the agent deletes something on a git repo, you can get it back. If it deletes it on your only copy, you can't.
- **Review before confirming.** Most agent tools show you what they're about to do before executing. Read that. Every time.
- **Scope credentials narrowly.** Don't give an agent your production database password when it only needs to read a config file.
- **Never give unsupervised access to production systems on day one.** Build trust incrementally, the same way you would with a new employee.

Critical: production = supervised

The rule is simple: if an action affects real users, real data, or real money, a human reviews it before it executes. Agent tools are powerful multipliers. The same quality that makes them fast at the right thing makes them fast at the wrong thing. Keep a human in the loop until you've verified the agent's judgement on your specific task through extensive supervised testing.

6. It doesn't know "now"

Every LLM has a **training cutoff**: a date at which its knowledge effectively stops. The model was trained on text collected up to that point, and it has no way to access anything that happened afterwards unless you tell it or a tool retrieves it.

This cutoff is often many months or even more than a year before you're using the model. The most capable models take a long time to train, evaluate, and release. By the time you're chatting with them, their "most recent" knowledge may already be outdated.

The practical implication: **don't ask a standard AI model for time-sensitive information** and trust the result. Current stock prices, election results, today's weather, the latest software versions, recently passed laws — these are all danger zones. The model will either honestly say it doesn't know (good) or fill in a plausible-sounding answer from its training data (bad and potentially outdated).

The workarounds:

- **Paste the current information into your prompt.** If you need the model to reason about a recent news article, copy the article text into your message. The model can analyse information you give it; it just can't fetch it independently.
- **Use a tool with web search enabled.** Claude.ai with search, Perplexity, the web-browsing version of ChatGPT — these retrieve current information before responding. They'll typically mention they're searching. Use them for time-sensitive questions.

7. It's not a professional

This is the limitation that most often comes up in advice-seeking conversations, and the one most people underestimate because the AI's tone is so convincing.

AI is not a doctor, a lawyer, or a financial advisor. It doesn't have a licence. It hasn't examined you. It hasn't read your contract, your medical history, or your tax situation. It knows general patterns from training data. Your situation is specific.

The gap matters most when consequences are high. An AI might describe the typical symptoms of appendicitis correctly but have no way to tell you whether *your* stomach pain is appendicitis, gas, or something else entirely. A lawyer might correctly summarise what a contract clause generally means but not know that in your jurisdiction there's a recent case that changes the interpretation entirely.

The right way to use AI with professional questions: use it to **understand the landscape** and **prepare better questions** for a real professional. "Explain what a non-compete clause is and what I should ask a lawyer about mine" is a great use. "Should I sign this non-compete?" is not one the AI can safely answer.

The "prepare your questions" superpower

AI genuinely shines at turning a scary professional appointment into a productive one. Ask it to explain the terminology you'll encounter, list the questions you should ask your doctor about a diagnosis, or summarise what a financial product is so you can have an informed conversation with an advisor. You'll get more out of the professional meeting, and you'll be a better-informed participant in your own decisions.

Putting it all together: the Pre-flight Checklist

The seven topics above boil down to a set of habits. The checklist below covers everything you should verify before shipping any AI-powered project — a script, an automation, an app, or even a regular workflow. Your state is saved between visits, so you can come back and tick things off as you set them up.

All 12 items must be checked to earn the green banner.

INTERACTIVE 4 OF 4 — PRE-FLIGHT SAFETY CHECKLIST

0 / 12 checked

API KEY & CREDENTIALS SAFETY

- ☐ **API key in environment variable**, not hardcoded in source files or pasted into chat.
- ☐ **Spending limit set** in my API console. I will not be surprised by a large bill from a runaway loop.

PRIVACY REVIEW

- ☐ **No passwords, tokens, or API keys** in any prompt or input that goes to a cloud model.
- ☐ **No unredacted medical, financial, or other people's private data** sent to a cloud service. If this data is involved, I use a local model.

HALLUCINATION VERIFICATION HABIT

- ☐ **I have verified any specific facts, citations, or statistics** the AI produced against an independent source before using them.
- ☐ **My prompts include "If you're not sure, say so"** wherever accuracy matters.

COST & SCALE CONTROLS

- ☐ **I tested with a small sample (5–10 items)** and confirmed cost per item before running any large batch.
- ☐ **Any loop or agent that calls an API has a circuit breaker or counter** that stops it if it runs unexpectedly long.

AGENT & AUTOMATION GUARDRAILS

- ☐ **Agent actions that affect real data or real users are reviewed by a human** before they execute.
- ☐ **I am not running agents against production systems** without extensive supervised testing on a safe equivalent first.

KNOWLEDGE & PROFESSIONAL LIMITS

- ☐ **For time-sensitive questions, I use web search** or paste current info directly — I do not rely on a model's training data for current events.
- ☐ **For medical, legal, or financial decisions, I consult a qualified professional.** AI has helped me prepare; a human expert makes the call.

State is saved in your browser. Come back and check things off as you set them up for a new project.

Summary

Seven risks, seven habits. None of them require you to become an AI safety researcher. They require building a mental checklist you run before trusting output, before pasting

data, before deploying a loop, and before making decisions that affect people.

The goal isn't to make you afraid of AI — it's to make you the kind of person who uses it well for years without getting burned. The people who get the most out of AI tools are not the ones who trust blindly or distrust completely. They're the ones who know the failure modes and work around them.

The homework page gives you three tasks that practice these habits on real scenarios. They're the most important exercises in this course because they build the instincts that protect you in everything that comes after.