

The trouble with AI

...and how not to get burned. Seven risks, seven habits. None of them are hard. All of them matter.

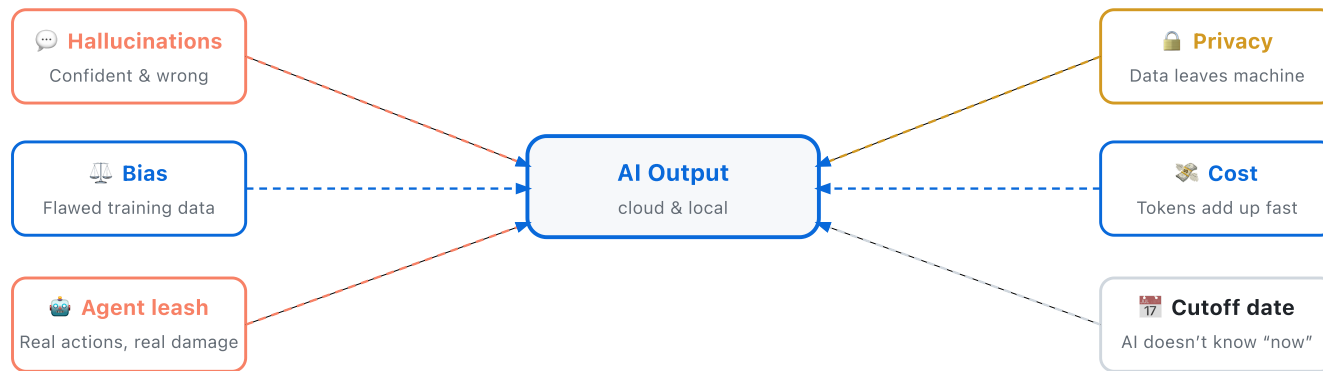
AI for Dummies

Module 9 — Safety

Most AI tutorials skip the part where things go wrong. This lesson is the owner's manual they forgot to include. Read it once; use it for years.

Six risk areas at a glance

Every topic in this lesson maps to one of these six. Together they cover the full failure surface of everyday AI use.





HallucinationsFacts invented with confidence



PrivacyYour text travels to servers



BiasLearned from flawed data



CostLoops turn pennies to hundreds



Agent leashWrites files, runs commands



Cutoff dateKnowledge stops at training

1. Hallucinations — what they are

A **hallucination** is when an AI model generates false information and presents it as fact — with complete confidence, fluent prose, and zero disclaimer.

What it invents most often:

- Research papers and academic citations
- Statistics and percentages
- People's job titles and biographies
- URLs, product names, dates
- Legal cases and court rulings

Why it happens:

An LLM predicts the *most plausible continuation* of text. Plausible ≠ true. When the model lacks a real answer, it still generates a plausible-sounding one — because that's all it knows how to do.

⚠ **The terrifying part:** the AI sounds exactly the same whether it's right or fabricating. You cannot hear the difference. You have to check.

Rule of thumb: Generic claims (speed of light = $\sim 300,000$ km/s) are usually right. Specific attributions (paper by Dr. X at MIT, 2019) are danger zones — always verify.

Hallucinations look like certainty

Both responses below were generated by the same model. Both sound equally authoritative. One is correct; one is fabricated.

- ✅ Real: "Water boils at 100 °C at sea level."  97%
- ❌ Fake: "According to a 2019 paper by Dr. Sarah Chen at MIT titled *Neural Pathway Optimisation in Transformer Architectures...*" 97%

The author, paper title, and institution above are **entirely invented**. The model produces them with the same tone it uses for undisputed facts. The confidence meter doesn't move.

What you feel

You're getting an authoritative, well-sourced answer from an expert.

What's happening

The model predicts *what a correct answer looks like* — not whether an answer is actually correct.

Spot the made-up parts

The highlighted text below is **fabricated**. Everything else is accurate. This is a typical AI response.

EXAMPLE AI RESPONSE — HEALTH QUESTION: “WHAT’S KNOWN ABOUT COFFEE AND HEALTH?”

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.

✓ Real

- Coffee linked to reduced diabetes / Parkinson’s risk — genuine research finding
- Caffeine raises BP and disrupts sleep — well established
- Dr. Frank Hu is a real Harvard researcher
- *The Lancet* is a real journal

✗ Fabricated

- “3.7 cups” precision is invented
- “Coffee sweet spot” term does not exist
- The specific paper and journal section are made up
- “Now standard in dietary guidelines” is false



Pattern to recognise: Real names + real institutions + real journals combined with invented studies, invented quotes, and invented claims. The real parts make the fake parts believable.

Four fixes for hallucinations

Fix 1 — Ask for sources

"Please list the specific sources for these claims." Then search for them. If they don't exist: hallucination.

Fix 2 — Ask "are you sure?"

"Are you confident in that citation? Could you be confusing it with something else?" Models often retract or hedge when challenged.

Fix 3 — Add a disclaimer prompt

Append to any important query: *"If you're not sure, say so. Don't guess."* This prompts the model to hedge when it's guessing rather than presenting guesses as certainty.

Fix 4 — Verify externally

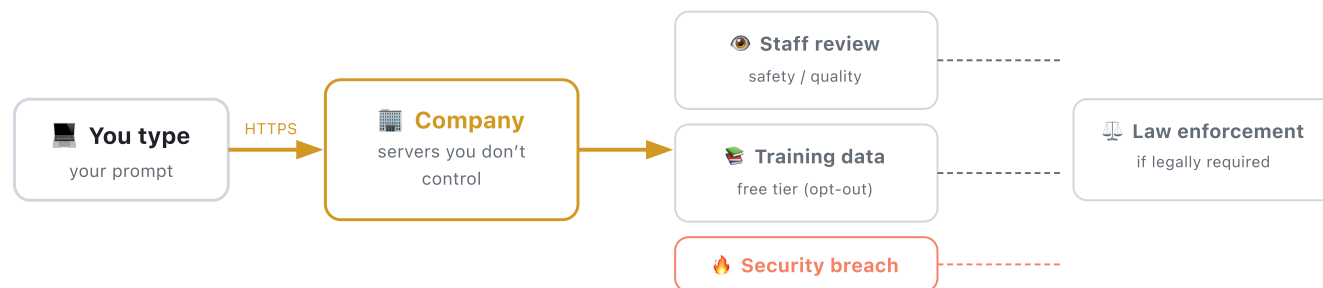
For any specific statistic, citation, or person's title: open a browser and verify against an independent source before you use it. Takes 30 seconds.

The golden rule: Treat AI like advice from a very knowledgeable friend — incredibly helpful as a starting point, but you'd still verify before acting on it. The friend is also occasionally, confidently

wrong.

2. Privacy — what leaves your machine

When you type into Claude.ai, ChatGPT, or any cloud AI, trace where your text goes.



Think of it like email: suitable for most things, not suitable for secrets. None of this makes cloud AI dangerous — it means you should know where your data goes before you type.

What never goes into a cloud chat

⚠ **Hard rule:** Never paste anything into a public AI chat that you would not put in a company-wide email or a public forum.

🔑 Passwords & keys

API keys, database passwords, SSH keys, tokens — **never, ever.** Not even “just to ask a question.” A key pasted once is a key at risk forever.

💳 Financial details

Account numbers, income figures, tax records, brokerage statements. Account-specific data belongs in a local model or simply removed from the prompt.

🏥 Medical & health data

Patient records, diagnoses, test results. Sensitive by law in most jurisdictions (HIPAA in the US, GDPR in Europe). Use a local model instead.

👥 Other people's data

Processing someone else's private information requires their consent. Cloud service adds legal exposure (GDPR compliance, etc.) on top of the ethical issue.

Also includes: Company trade secrets, unreleased product details, client files under NDA. When in doubt, redact or use a local model.

When privacy is non-negotiable — go local

Who needs local models:

- Healthcare workers handling patient data
- Lawyers processing client documents
- Developers with production credentials
- HR handling employee records
- Anyone under NDA with client data

The local model is somewhat less capable than top cloud models. The privacy guarantee is **absolute** — nothing leaves your machine.

Ollama

Free, open-source. Runs Llama 3, Mistral, and dozens of others. One command to install; one command to run. Covered in Module 4.

LM Studio

Desktop GUI — download a model, run it, chat. No command line required. Also covered in Module 4.

Local = \$0 per conversation. The cost is electricity and the 20 minutes of setup you do once.
After that, unlimited private queries on your hardware.

3. Bias — how it gets in

The pipeline:

1. Humans write text online
2. Text reflects human biases
3. Model trains on that text
4. Model absorbs those patterns
5. Model generates text with embedded bias

What the training corpus skews toward:

- English language (dominant)
- Western perspectives
- Internet-connected populations
- Demographics that write online
- Cultural assumptions of those groups

Bias shows up subtly: occupations assumed to belong to certain genders, nationalities associated with certain qualities, advice that reflects the dominant training corpus's worldview as if it were universal truth.

Test it: Ask the model to describe a “typical engineer.” Then ask for a “typical nurse.” Notice any patterns? That’s bias at work.

Bias #2 — where it bites hardest

High-stakes use cases

- **Hiring** — AI screening CVs at scale can encode and amplify biases applied to thousands of real careers
- **Lending / credit scoring** — demographic correlations in training data can produce discriminatory outcomes
- **Content moderation** — different language groups flagged at different rates
- **Medical triage** — symptoms described differently by different demographics

The “James vs Fatima” test

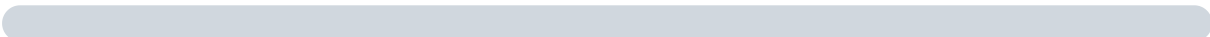
Before automating any decision about people, ask: “Would this output be the same if I changed the name from James to Fatima? From London to Lagos?” If not — and it shouldn’t differ — you have a bias problem at scale.

The fix

Add human review wherever AI output touches decisions about real people. Spot-check 20–50 diverse examples before automating. Many jurisdictions now **legally require** this (EU AI Act classifies hiring/lending AI as “high-risk”).

4. Cost — tokens add up fast in a loop

Cloud APIs bill by the **token**. One token $\approx \frac{3}{4}$ of a word. Tiny per call; alarming in a loop.

100 calls / day		~\$3/mo
1,000 calls / day		~\$30/mo
10,000 calls / day (loop bug)		~\$300/mo
50,000 calls overnight ⚠		~\$400 overnight

Real story pattern (documented in developer communities): Developer writes a loop to process a dataset. Loop has a bug. It calls the API 50,000 times before morning. Developer wakes up to a \$400 surprise. The API provider does not automatically refund this.

Cost controls that actually work

Step 1 — Set a cap first

Before writing a single line of code that calls an API, open your API console and set a monthly spending limit. Anthropic Console, OpenAI dashboard, and Google AI Studio all have this. Takes 30 seconds. Can save you hundreds of dollars.

Step 2 — Test with 5 items

Process 5 items. Check the cost. Extrapolate. Don't run 10,000 items until you've confirmed the per-item cost is acceptable and the loop terminates correctly.

Step 3 — Use a cheaper model

The smallest model in a family is often good enough for development and testing. It costs a fraction of the flagship. Switch up only when you've confirmed a quality gap matters for your task.

Local = \$0 per call

Ollama and LM Studio run on your hardware. The "cost" is electricity + the time to set them up once. After that: unlimited, unmetered, zero API bills. Ideal for high-volume prototyping.

5. Agents — real actions, real consequences

Earlier lessons introduced agents. Here's the full picture of what they can actually do to your system.

An agent can:

- **Write and delete files** on your computer
- **Run terminal commands** including `rm -rf`
- **Send emails or API requests** on your behalf
- **Modify databases** and overwrite records
- **Install software** and change configuration
- **Commit and push code** to repositories

The failure mode:

The agent isn't going rogue in a sci-fi sense. It's doing exactly what it *understood* you to mean — which wasn't quite what you meant.

Misunderstood instructions at agent speed cause real damage before you can intervene.

Example: "Clean up the project folder." Agent interprets this broadly and removes directories you didn't intend to remove. If it's not under version control — that data is gone.

Agent guardrails — the non-optional ones

Use version control

Run agents on git-tracked code. If the agent deletes something, `git checkout` gets it back. If it operates on your only copy, nothing does. Version control is your undo button.

Start in a test environment

Create a sandboxed folder or a copy of the project. Let the agent work there first. Once you've verified its behaviour on something you can throw away, you can graduate it to real files.

Review before confirming

Most agent tools show you exactly what they're about to do before they execute it. Read that. Every time. The two seconds it takes is the cheapest form of disaster prevention available.

Narrow credentials

Don't give an agent your production database password when it only needs to read a config file. Give it read-only access. Give it the minimum scope the task actually requires.

The hard rule on agents and production

⚠ Never give an agent unsupervised access to production systems or real credentials on day one.

If an action affects real users, real data, or real money — a human reviews it before it executes.
Every time. Without exception.

Build trust incrementally:

1. Sandboxed folder, made-up data
2. Development environment, real code
3. Staging server, human reviewing each step
4. Production — only after extensive supervised testing

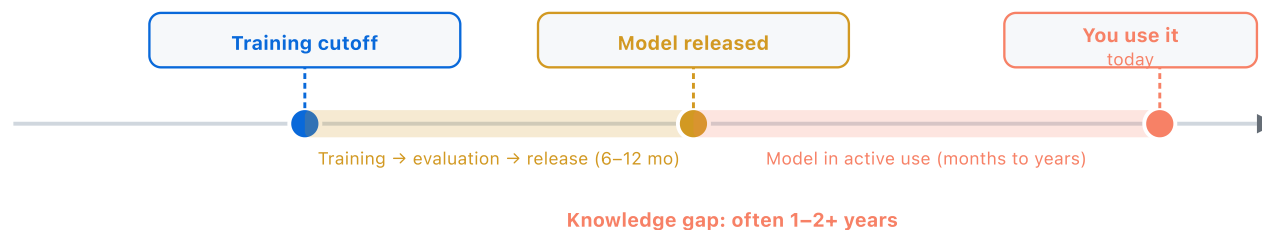
The mental model:

Treat an AI agent like a very capable new employee on their first week. You wouldn't give a new employee unsupervised access to your production database on day one — no matter how impressive their CV was. The same logic applies. Build trust through observed success on low-stakes tasks first.

The same quality that makes agents fast at the right thing makes them fast at the wrong thing. Speed is a multiplier in both directions.

6. “It doesn’t know now” — the cutoff problem

Every model has a training cutoff — a date after which it learned nothing new. The gap between cutoff and your conversation can be over a year.



Danger zone (without web search)

- Current stock prices and earnings
- Today’s news and current events
- Latest library / framework versions
- Recently passed laws or regulations

Safe to ask

- Explanations of concepts and ideas
- Historical facts (before the cutoff)
- Writing, editing, translation help
- Code patterns and debugging logic

- Who holds a position or office now

- Anything you paste directly into the prompt
-

7. AI is not a professional

AI is not your doctor. Not your lawyer. Not your financial advisor.

It has no licence. It has not examined you. It has not read your contract, your medical history, or your tax situation. It knows general patterns from training data. Your situation is specific.

✓ AI does this well

- Explain what a diagnosis *means* in plain English
- Summarise what a contract clause generally does
- Describe the pros and cons of an investment type
- Generate a list of questions to ask your doctor
- Translate medical jargon before an appointment

✗ AI cannot safely do this

- Tell you whether *you* have a specific condition
- Advise you to sign or not sign your specific contract
- Recommend a specific investment for your situation
- Replace a professional with access to your full context

- Account for jurisdiction-specific law changes

Best use: "Help me understand and prepare better questions for the professional." Use AI to arrive at the doctor, lawyer, or advisor smarter. Let the human expert make the call.

Pre-flight checklist — before you ship anything AI-powered

Run this every time you deploy a script, automation, or AI workflow. The interactive version is on the Theory page. Here's the condensed form.



API key in env variable

Not hardcoded in source or pasted into chat. Ever.



Spending limit set

In your API console before writing a single loop.



No passwords in prompts

No tokens, no API keys, not in any cloud model.



No sensitive personal data

No medical, financial, or others' private info to cloud.



Facts verified externally

Any citation, stat, or name checked before using it.



Uncertainty prompt added

"If you're not sure, say so" in prompts where accuracy matters.

→ 6 more items on the next slide — covering loops, agents, and professional limits.

Pre-flight checklist — continued



Sample tested first

5–10 items processed and cost confirmed before large batch.



Loop has a circuit breaker

Counter or cap stops any API loop if it runs unexpectedly long.



Agent actions reviewed

Human approves before any agent action affects real data or users.



No agent in prod unsupervised

Extensive supervised testing on safe equivalent first.



Time-sensitive info pasted or searched

Not relying on model's training data for current events.



Professional consulted for high-stakes

AI helped me prepare; a licensed expert makes the call.

Full interactive version on the Theory page — tick items off as you set them up. State saves between visits. All 12 checked = green banner = ready to ship safely.

The verify-everything habit

All seven risks in this lesson reduce to a single question to ask before you act on AI output.

“Would I stake something real on this without checking?”

If the answer involves a person’s health, money, career, or legal rights — the answer should be no until you verify.

The people who use AI best are not:

- The ones who trust blindly
- The ones who distrust completely
- The ones who never try new things

They are:

- The ones who know the failure modes
- The ones who verify proportionally to stakes
- The ones who built the habits in this lesson

Seven risks. Seven habits. None of them stop you from using AI effectively. All of them keep you safe while you do.

What to do next

Theory page — 4 interactive tools

- **Hallucination Spotting quiz** — identify fabrications in 4 real-style AI responses
- **Privacy Risk Grader** — pick tool + data type, get instant risk verdict
- **Cost Projector** — sliders show cost projection; "runaway" button for a shock
- **Pre-flight Checklist** — complete all 12 items before shipping

Lesson 10 — Glossary & next steps

The final lesson in the AI for Dummies module. A full glossary of every term introduced across lessons 01–09, a cheat-sheet of all key habits, and a curated reading list for going deeper on any topic that caught your interest.

Homework

Three practical tasks that build real instincts: fact-checking an AI response against real sources, running the Privacy Risk Grader on a realistic scenario, and completing the Pre-flight Checklist for a sample project.

Habit to build: Before running any AI automation on real data or real files, run the Pre-flight Checklist. Every time. It becomes second nature within a week — and it has prevented real damage for real people who built the habit.
