

Lesson 07 — Real-world examples

Five categories. Prompts you can copy today — no coding required.

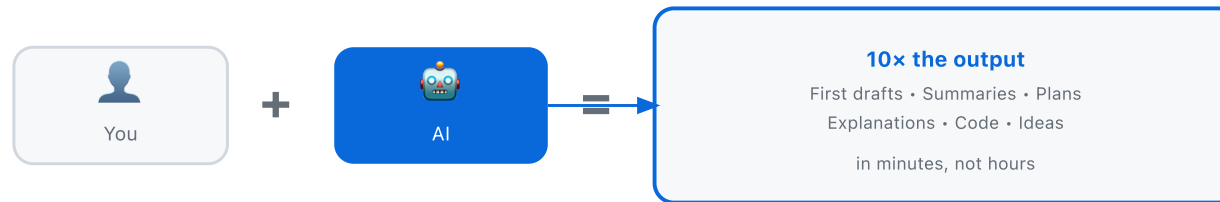
AI for Dummies

Module 7

This lesson is different. No theory — just **real prompts, real outputs, real transformations**. By the end you'll have a mental library you can reach for whenever you need it.

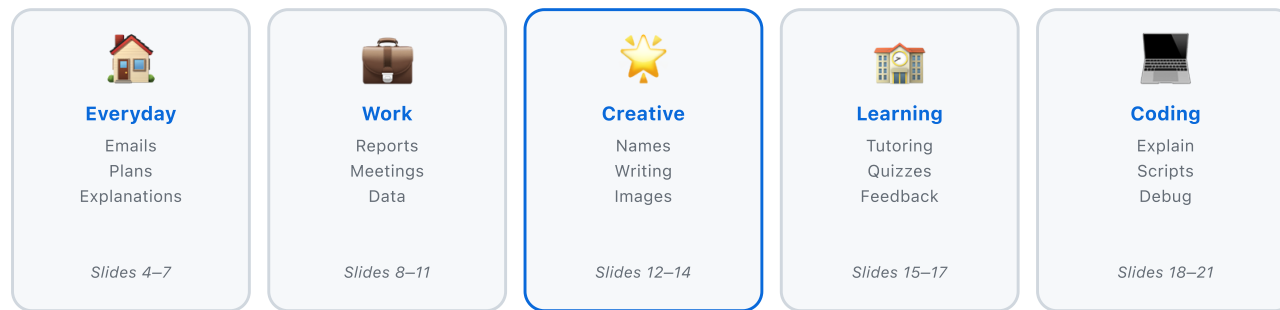
AI is a force multiplier

Same person. Same knowledge. Dramatically more output per hour.



The pattern across every example in this lesson: AI handles the first draft or the structured thinking — the hardest part to start. You steer, review, and decide.

Five categories — one lesson



Each section has a real prompt box you can copy. No theory — just examples.



Category 1 — Everyday life

The most underestimated category. These are the tasks we do every day — for family, home, and personal life — that consume real time and mental energy. AI handles all of them fast.

What belongs here:

- Writing emails you've been avoiding
- Summarising documents you don't want to read
- Planning trips, meals, workouts with constraints
- Understanding confusing things (contracts, medical reports, tax forms)

The pattern:

Take this *[messy input]*
and give me *[clean output]*
for *[my situation]*.

Messy in → polished out. That's the whole idea.

Everyday #1 — messy thoughts → polished email

BEFORE — YOUR RAW NOTES

"need to tell landlord roof is leaking again, last time took 3 weeks, really frustrated, want it fixed by friday but don't want to sound rude, they ignored my last message"

AFTER — AI OUTPUT

Subject: Roof leak — second notice

Hi David,

I'm writing again about the roof leak I reported three weeks ago. Unfortunately it persists and is getting worse. Could you confirm a repair date by Friday? I'd appreciate a quick response this time.

The prompt that did it:

Turn my notes into a polite but firm email to my landlord. Under 80 words. Flag the history (previous ignored message). Notes: [paste above]

Why this works: you gave the tone instruction (*polite but firm*), the constraint (*under 80 words*), and the context (*previous message ignored*). The AI threaded all three.

Thanks,
Maria

**Everyday #2 — read a 12-page contract in
30 seconds**

THE TASK

A 12-page rental agreement you received on a Friday evening. Dense legalese. You're moving in Monday.

AI OUTPUT — 5 THINGS TO KNOW

1. Notice period is **60 days** (not the standard 30).
2. Pets require written approval — not just verbal.
3. Landlord can enter with **24 hr notice**.
4. Deposit is non-refundable if lease breaks before 6 months.
5. ⚠️ Unusual clause: you pay for any plumbing repairs under €200.

The prompt:

Here is a rental contract. Give me the 5 most important things I need to know as a tenant, in plain English. Flag anything unusual or that I should be concerned about.

[paste document text here]

Caveat: use this to know *what to read carefully* in the original — not as a replacement for reading the key clauses yourself.

Everyday #3 — plan a trip around your real constraints

The prompt:

Plan a 5-day trip to Lisbon for 2 adults in late October. Budget: ~€1 500 total excluding flights. We like food, walking, and history. We hate tourist traps. Give a day-by-day outline with 2–3 activities per day and a rough daily cost.

SAMPLE AI OUTPUT (DAY 1)

Day 1 — Alfama & Fado

Morning: walk Alfama, skip the main square crowds — head to Miradouro da Graça instead. Afternoon: MAAT museum (€5). Evening: dinner at a neighbourhood tasca in Mouraria, followed by a live fado show (~€35 pp). *Est. day cost: €80 for two.*

Works the same way for:

- Weekly meal plan (with dietary constraints)
- Workout schedule (with equipment limits)
- Budget for a home renovation
- Study plan for an exam

Constraints make better prompts. €1 500 total, hate tourist

traps is ten times more useful than "plan a Lisbon trip."

Everyday #4 — same topic, two levels

Ask for the simple version, then the expert version. Same AI, same conversation.

SIMPLE VERSION

Explain compound interest like I'm 10 years old, with a simple analogy.

"Imagine you put €100 in a magic piggy bank. Every year it adds 10% of whatever is inside — not just the original €100. So after year 1 you have €110, year 2 you have €121, and it snowballs. Money making more money."

EXPERT VERSION (FOLLOW-UP)

Now explain it for a finance professional. Give the formula and a worked example with monthly compounding.

$A = P(1 + r/n)^{(nt)}$. At $P=€10\,000$, $r=6\%$, $n=12$, $t=10$:
 $A = €10\,000 \times (1.005)^{120} = \textbf{€18\,194}$. The monthly compounding adds ~€600 vs annual compounding over the decade.

Try this pattern on anything confusing — a news story, a contract clause, a medical term, a physics concept. Same question, two very different explanations, back to back.



Category 2 — Work

Knowledge workers spend a large fraction of their day on writing, meetings, and communication. These are not the core of most jobs — they're the scaffolding around it. AI handles scaffolding well.

High-value work prompts:

- Meeting notes → action items + owners
- Bullet points → executive report
- Data table → trend analysis
- Technical text → customer-facing copy

The prompt pattern:

Here is *[raw material]*.
Turn it into *[structured output]*
for *[audience]* in *[tone/format]*.

Work #1 — meeting notes → action list

BEFORE — RAW NOTES (45-MIN MEETING)

"discussed new landing page - tom said he'd get designs by eow. budget still unclear. marketing wants analytics set up before launch. dev says 2 weeks if designs on time. need to confirm launch date maybe 15th? sarah to check legal about disclaimer. follow up on email list migration - was supposed to be done last week."

AFTER — STRUCTURED OUTPUT

Action items:

- Tom — Landing page designs by **Fri EOW**
- Sarah — Legal check on disclaimer copy (*no deadline set*)
- Dev team — 2-week build once designs received
- [Unassigned] — Confirm launch date (~15th)
- [Unassigned] — Email list migration (*overdue!*)

Open questions: Budget unconfirmed. Analytics owner?

Here are my raw meeting notes. Extract all action items with owner + deadline (or "unassigned"). Also list open questions. Notes: [paste]

Work #2 — draft a report from notes

The prompt:

Write a one-page Q1 sales summary report. Key facts: revenue €420k (up 12% vs Q4), top product was the Pro plan, biggest challenge was supply chain delays in March. Audience: executive team. Tone: professional, direct. Include a 3-bullet highlights section at the top.

MOCK AI OUTPUT

Q1 2026 Sales Summary

Highlights:

- Revenue: **€420k** — up 12% quarter-over-quarter
- Pro plan drove 58% of new revenue
- March supply delays cost est. €30k in deferred deals

Performance: Q1 exceeded Q4 across all key metrics. The Pro plan outperformed projections by 18%...

Risk: Supply chain vulnerability remains unresolved. Recommend a contingency review before Q2.

The blank page problem disappears. Give AI the skeleton; you add the company-specific numbers, context, and political nuance. Takes 5 minutes, not 2 hours.

Work #3 — ask questions about a spreadsheet

The prompt:

Here is a table of monthly sales by product for the past 12 months. What are the three biggest trends? Which product is growing fastest? Is anything declining? [paste table]

What a pasted table might look like, and what AI extracts:

Month	Pro	Starter	Enterprise
Jan	€38k	€21k	€15k
Feb	€42k	€20k	€17k
Mar	€45k	€18k	€19k
Apr	€51k	€17k	€22k

AI analysis:

- Pro: **+34% over 4 months**, fastest grower
- Enterprise: **+47% growth**, accelerating
- Starter: **-19% trend**, declining — investigate

Work #4 — adapt content for a different audience

TECHNICAL ORIGINAL

"The new webhook integration enables real-time event-driven architecture by pushing POST requests to a configured endpoint whenever state transitions occur. Supports HMAC-SHA256 signature verification. Retry logic uses exponential backoff up to 5 attempts."

Rewrite this for our non-technical sales team. No jargon. Focus on what the customer can do and why it helps them. Under 60 words. Friendly tone.
Original: [paste]

CUSTOMER-FACING OUTPUT

New: instant updates to your other tools

Connect your account so your CRM, Slack, or spreadsheet updates the moment something changes — no more manual exports or waiting overnight. Fast, secure, and if your system is briefly offline, we automatically retry for you.

Also works in reverse: translate customer feedback into a technical bug report. Or adapt one document for five different audiences in minutes.

Category 3 — Creative & brainstorming

AI doesn't replace your creativity. It **unsticks you** when you have nothing, and **gives you options** when you have too many possibilities to start. Your taste is still the filter.

The prompt: give me 10 names for a dog-walking app. Short, memorable, playful. Avoid "paw" or "wag" — taken. Mix single words and short phrases.

Barkly

Walkr

Fetch & Go

Sniff Club

Roamio

TailTime

GoodBoy

Strut

LeashUp

RunRover

The trick: your taste steers the process. "I like #1 and #5. Give me 10 more in that direction." You're not accepting output — you're curating it.

Creative — getting unstuck on writing

Stuck on a paragraph? Ask for three different versions to react to. Pick one. Polish it.

The prompt:

I'm writing a travel blog about Kyoto in autumn. Stuck on the opening paragraph. Give me three very different options: one lyrical and atmospheric, one punchy/direct, one that opens with a specific anecdote. I'll choose the one that fits my voice.

LYRICAL OPTION

"The maple leaves come down in Kyoto like a slow red argument with gravity — each one holding on a moment longer than it should before letting go."

PUNCHY OPTION

"Everyone tells you Kyoto is beautiful in autumn. They're not wrong. But they don't tell you how crowded it is at the famous spots, or where to go instead."

Key mindset: you often discard all three — but seeing the contrast tells you what you *actually* want to write. That's the value.

Creative — generating images with text

Tools for this:

- **Midjourney** — highest quality, Discord-based
- **DALL•E 3** — inside ChatGPT, easy to use
- **Adobe Firefly** — good for commercial use
- **Stable Diffusion** — free, runs locally

Good for: blog illustrations, social post images, presentation visuals, logo brainstorm, product mockups, poster ideas.

Example image prompt:

A flat-style illustration of a cozy home office at night, warm lamp light, plants on the desk, minimal and modern. Muted color palette. No people.

Prompting images follows the same rules as text: be specific about style, mood, and what to exclude. "Flat-style, muted palette, no people" is a complete brief.



Category 4 — Learning: the 2 a.m. tutor

A patient tutor for any subject, available at any hour, that adapts to your exact level and never makes you feel stupid for asking again. This is the most powerful use case for serious learners.

The setup prompt:

You are a patient tutor helping me learn basic economics. I'm a complete beginner. Start with supply and demand — explain it simply with a real-world example, check that I understand, and only move on when I say I'm ready.

What you get:

- A plain-English explanation at your level
- A check-in: "does that make sense?"
- Deeper content when you ask for it
- Unlimited "try a different analogy" rounds
- Moves at your pace, not a class schedule

Human tutors run out of analogies. AI doesn't.

Learning — quiz yourself on anything

Reading is passive. Active recall — being tested — is how things stick. Research-backed, easy to set up.

The prompt:

```
Ask me 10 multiple-choice questions on the French Revolution. Give me one question at a time. After I answer, tell me if I'm right and explain briefly why. Keep a running score. Go.
```

Sample Q&A exchange:

AI: Q1/10 — What year did the French Revolution begin?

A) 1776 B) 1789 C) 1799 D) 1804

You: B

AI: Correct! 1789. The storming of the Bastille on July 14th is the symbolic start. Score: 1/1. Next question...

Works for any subject: history, science, coding languages, foreign vocabulary, material from a course you're enrolled in. You can even paste in the content: "Based on this chapter, create 8 short-answer questions."

Learning — feedback on your writing

The prompt:

Here is my essay draft. Give me honest, structured feedback on three things: 1) strength of the argument, 2) clarity of the structure, 3) writing quality. Don't rewrite it — just tell me where it's strong and where it needs work.

[paste essay]

AI feedback example:

Argument (7/10): Your thesis is clear, but paragraph 3 contradicts the claim in paragraph 1 without acknowledging the tension.

Structure (6/10): The conclusion introduces a new idea that should appear earlier. Consider moving it to the second section.

Writing (8/10): Clear and readable. Watch out for passive voice in the opening — it weakens the authority of your first impression.

This kind of structured feedback — normally requiring a mentor or writing group — is now available any time for free. Same works for code, business plans, or any reasoning exercise.



Category 5 — Coding (even for near-beginners)

You do *not* have to be a developer. The best coding use cases are often for people who know a *little* code — enough to have a problem, not enough to solve it alone.

What near-beginners use it for:

- Explain what a piece of code does, line by line
- Write small scripts for repetitive tasks
- Debug: paste an error, get the fix + explanation
- Learn: “explain what a for-loop does, 3 examples”

The ask-for-explanation habit:

Always add “explain each line” or “explain why” to any coding prompt. You get the script *and* you understand it. That’s how near-beginners actually level up.

Coding #1 — write me a script

The prompt:

Write a Python script that goes through all files in a folder and renames any .jpeg file to .jpg. I'm on Mac. Keep it simple and explain each line.

AI OUTPUT

```
import os

folder = "/Users/you/photos"  # ← change this

for filename in os.listdir(folder):
    if filename.endswith(".jpeg"):
        old = os.path.join(folder, filename)
        new = old[:-5] + ".jpg"
        os.rename(old, new)
        print(f"Renamed: {filename}")
```

After you get this, ask follow-ups: "What if I want it to include subfolders?" or "Can it ask for confirmation first?" The conversation continues until it does exactly what you need.

Coding #2 — explain this code line by line

Inherited mystery code:

```
def process(data):  
    result = {}  
    for item in data:  
        k = item['id']  
        result[k] = result.get(k, 0) + 1  
    return {k: v for k, v  
            in result.items()  
            if v > 1}
```

Explain what this code does. Go line by line for anything complicated. Assume I know Python basics but not advanced concepts.

AI EXPLANATION

What it does overall: finds duplicates. It counts how many times each `id` appears in the list, then returns only the ones that appear more than once.

Line by line:

- `result = {}` — empty dictionary to store counts
- `result.get(k, 0) + 1` — if key exists add 1, otherwise start at 0
- Last 3 lines — dictionary comprehension: keep only entries where count > 1

Coding #3 — find and fix the bug (with the WHY)

The prompt:

I got this error running my script. Tell me what caused it and how to fix it. Explain why it broke so I understand the concept.

Error:

`TypeError: can only concatenate str (not "int") to str`

Code:

```
age = 25
print("I am " + age + " years old")
```

AI RESPONSE

What caused it: Python won't let you join a string and an integer with `+`. `"I am "` is a string; `25` is a number. They're different types.

The fix — two options:

```
# Option 1: convert int to string
print("I am " + str(age) + " years old")

# Option 2: f-string (cleaner)
print(f"I am {age} years old")
```

The “explain why it broke” instruction is the key. You understand the concept for next time — not just copy-pasted code you don't understand.

The pattern across all five categories

AI is a force multiplier for getting started. In every single example in this lesson, the AI did one of two things: it handled the *first draft*, or it provided the *structured thinking* that was blocking you. You steered, reviewed, and decided.

What AI is great at:

- First drafts of anything written
- Transforming messy input into clean output
- Generating options for you to react to
- Explaining things at the right level
- Writing small code tasks from a description

What you still do:

- Provide the real context and constraints
- Verify anything consequential
- Make the final call
- Polish what doesn't sound like you
- Know when the output is wrong

Honest limitations — where AI falls short

Use with caution:

- **Real-time facts** — AI doesn't know today's news or live prices. Always verify.
- **Precise arithmetic** — AI can get numbers wrong. Double-check calculations.
- **Niche expertise** — obscure legal, medical, or technical edge cases. Starting point, not final authority.

The simple rule:

If being wrong has real consequences — health, money, legal, sending to customers — treat AI as a **research assistant**, not a final authority. Use it to get oriented fast, then verify what matters.

The people who get the most value from AI are the ones who *also* know when not to trust it.

Use it on something real this week

Everything in this lesson becomes real skill the moment you use it on an **actual task from your life**. Not a practice exercise — a real email you've been avoiding, a document in your inbox, a piece of writing you're stuck on.

Pick one this week:

- Draft an email you've been putting off
- Summarise a document sitting in your inbox
- Plan a trip, meal, or project with constraints
- Get help preparing for a meeting

Or:

- Ask for an explanation of something confusing
- Quiz yourself on something you're learning
- Name a project or brainstorm a list
- Paste an error message and ask for the fix

Theory becomes skill at the moment of first use on something real.

What's coming next

Lesson 08 — AI in your own projects

From scripts to small apps. Cloud vs local. How to fold AI into something you're building — even if you're not a developer.

Lesson 09 — Safety, privacy & cost

What AI can't do. Where it goes wrong. How to stay protected and in control.

Before you go:

Read the Theory page for the full interactive use-case explorer (30+ prompts), the before/after gallery, and the pattern recognizer quiz. Each prompt is copy-ready.

The five prompt patterns in this lesson — transform / structure / generate-options / tutor-dialogue / write-and-explain — cover 90% of what most people will ever need.

