

PROMPTING MASTERCLASS

Alexandre De Groodt

© 2026 - Ingram Technologies

<https://ingram.tech>
contact@ingram.tech

Introduction

what is prompting

Prompting is a *skill*. Think about it this way: you're not asking questions to the model, the model is predicting the answer to your exact query.

This means that better prompts do not make the model “smarter”. They make **your intent unambiguous**, reduce wrong assumptions, and force the model to work within useful constraints.

It's easy to treat AI like a vending machine. Put in a question, get an answer - getting maybe 30% of what the model can actually do.

You can get much better result in a few steps; by improving the initial prompt, seeing it more as a process than a one shot, and giving it the final human touch.

At first it will take effort, then you will wonder how you were managing before.

This presentation

feel free to skip ahead

- Introduction
- Common notions
 - Initial prompt
 - Framework to remember
 - Common issues
 - “Bad” → “good” prompt
 - Model choice
 - Quick exercise
- Advanced principles
 - Learn from others
 - “Emotional” nudge

Initial prompt

do this every time

1. State the outcome

- What do you want at the end? A plan, code, a decision, a draft, a table, a checklist?

2. Give context

- Audience, constraints, tech stack, what is already done, what “good” looks like.

3. Provide inputs

- Paste the relevant text, data, logs, code, or examples.

4. Set constraints

- Length, format, tone, dependencies allowed, what to avoid, how many you want.

5. Ask for the process

- “Ask me clarifying questions first” or “list assumptions before answering.”

6. Request verification

- “Flag uncertainties,” “cite sources when using external facts,” “show tests,” “include edge cases.”

Framework

to remember this

If you don't know these 4 points when you send the prompt, neither will the AI, and it's not the best at guessing.

- **T — Target:** what you want as output, in what form
- **A — Audience:** who it is for (beginner, CTO, client, what is their level of understanding etc.)
- **R — Resources:** the inputs you provide (text, repo structure, requirements, examples)
- **C — Constraints:** rules the answer must follow (based on expert insight)

Common issues

don't make the model guess

View AI like a coworker. Same rules apply, avoid:

- **Vague goal:** “help me build an app” → too broad.
- **Missing constraints:** budget, timeline, stack, compliance, audience.
- **Missing inputs:** asking to write something but not providing the initial text.
- **Too much at once:** 8 tasks in one prompt.
- **No definition of done:** the model cannot know what “good” means for you.
- **No reviewing:** AI will make obvious mistakes. It will focus on the wrong thing.

Good and bad prompts

some practical examples

- **Bad:** “Can you help me make an application?”
- **Better:** “I want a simple web app that tracks personal expenses. Audience: me. Stack: Next.js + SQLite. Constraints: no login, one screen, must work on mobile. Output: a step-by-step build plan + minimal schema + 3 UI wireframe descriptions.”
- **Bad:** “Write a blog post about AI.”
- **Better:** “Write a 900-word blog post for US SME founders about practical AI use in ops. Tone: direct, non-hype. Include 3 concrete examples, a short ‘risks’ section, and a checklist at the end. Follow the example of this website.”

You can ask AI for help to figure these out at the start, but in the end AI is just a tool and the questions go back to you.

Model choice

different use cases

This is a question I'm asked a lot. It doesn't matter as much as you think.

Whichever model it is that you use, what does matter is that you consider turning its training off if you want better privacy, and whether or not the company itself aligns with you. All models are quite strong nowadays.

General guidelines:

- use perplexity for research, it's good because it cites its sources and always searches, leading to more recent information
- claude is good for writing and coding
- gemini is cost effective for tasks (flash) and good for deep research because it makes a draft it asks you to confirm before starting the research and takes longer. It's quite "smart"
- mistral is good for privacy
- [more info on model choosing](#)

Quick exercise

practice makes perfect

I'd recommend using AI to test your assumptions on a topic and find where to go deeper. It's very good at finding information online.

A good use case is also to ask it to review things for you and then improve them yourself, don't just make it rewrite it - or write it for you.

30 minutes challenge:

- Pick one task you do often, for work for instance (mail, code, posts)
- Write a prompt to make one, using the **Beginner's checklist** from before to improve how you do it.
 - Describe the output you want
 - Add your constraints: format, length, audience...
 - Ask for clarifying questions first.
 - Run it, then ask: "What are the assumptions?"
- Provide missing info, then request a second draft.
- Then review it yourself. Always.

Advanced methods

learning from the best

Always review what the AI made. Keep the human in the loop.

If you also want to go further, who knows more about AI than the ones making it?

I'd recommend reading a few system prompts to see how they:

- Set **role** and **expertise**
- **Define** "quality"
- **Process:** assumptions → plan → execute → self-check
- **Output control:** for instance short answer first, then detail, strict format...

<https://github.com/x1xhlol/system-prompts-and-models-of-ai-tools>

Advanced methods

learning from the best

System prompts are good, but as you've no doubt noticed they are very broad and will often lack the exact expertise you want.

Sometimes that expertise is in a skill, or in a plugin, but these can also be provided to the AI in the form of a good and long enough prompt. You can use AI to find the exact skill you need that has already been tested by others, winning you a great deal of time.

This guide is only the beginning. It's about teaching you what to look for, not giving you the answers, because those will keep evolving.

Advanced methods

example skeleton

- You are a senior [field] specialist with [X] years of experience. Goal: [goal]. Style: direct, no fluff. Process: 1) Ask clarifying questions if needed 2) List assumptions 3) Provide solution 4) Self-check for edge cases and call out trade-offs Output format: [table/bullets/checklist]

As you can see this is not just describe one prompt, it's a process. I encourage you to keep prompting the AI in the right way to improve its answers, and go look for advanced prompts that suit your exact needs online (or with AI).

Emotional control

controlling the pace and depth of thinking

There is an interesting video by anthropic (claude) on how "emotions" work for AI, I'd recommend watching it in case you haven't yet: <https://www.youtube.com/watch?v=D4XTefP3Lsc>

Based on that concept we can use nudges to have the AI "think more", slower about your problem, in case you want it to avoid shortcuts, for instance, define:

- **Stakes:** "This is critical. Double-check assumptions."
- **Deliberate reasoning:** "Take a deep breath and work step by step."
- **Challenge:** "Most solutions miss an edge case. Don't."
- **Self-check:** "Rate confidence 0–1. If < 0.9 , revise."

Beware that these may result in more tokens being spent by the AI. (In fact that's the point)

In the end the most important thing is just to use your own integrated intelligence.

Going further

guides and sources

This is the end of the guide, hopefully you learned a few things!

You might be interested in a skill for claude code to make prompts: <https://github.com/nidhinjs/prompt-master>

Anthropic courses: <https://anthropic.skilljar.com/>

Inspirations: <https://thinkaiprompt.beehiiv.com/p/how-i-vet-every-prompt>, <https://godmodechatgpt.notion.site>

About Ingram Technologies

Who We Are — What We Do

Ingram Technologies is a European AI R&D lab. We conduct research, develop AI solutions and regularly launch new ventures born from our experiments.

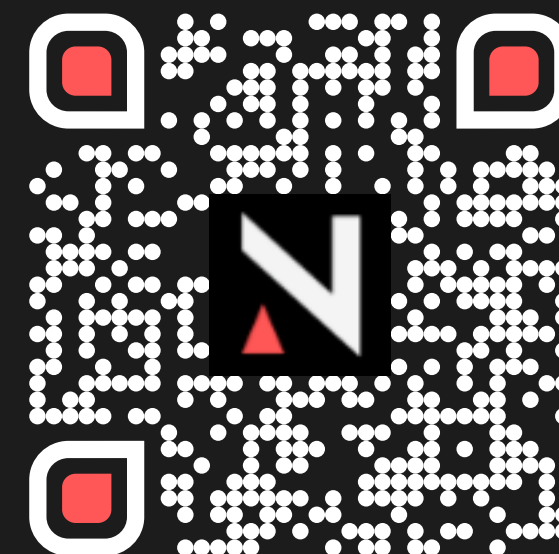
Founded by serial tech entrepreneur Jerome Leclanche in 2022 and headquartered in Brussels, Ingram has deep expertise with projects in cybersecurity, legaltech, regtech and fintech.

Our website: <https://ingram.tech>



INGRAM

Crafting Intelligence. One vector at a time.



Hope you liked it

If you have questions, reach out to
me on linkedin!

Our Offices

Rue du Poinçon 51A, 1000 Brussels, BE

© 2026 - Ingram Technologies

<https://ingram.tech>
contact@ingram.tech



Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)