

01 / METHOD

Free prompt engineering toolkit

An editable template, a response checklist and three explained examples. Short resources to keep at hand when using AI at work.

One request, four blocks

Before sending a prompt, define the result you need and how you will judge a useful answer. Tags separate instructions and materials: you do not need to know XML to start.

Goal

Describe the result, rather than just the topic. “An email to agree a new delivery date” is more specific than “a professional email”.

Method

Specify useful steps and what to do when information is missing. Ask for criteria and a short, checkable explanation, rather than a reconstruction of the model’s internal reasoning.

Attachments

Include relevant material only. If uploading a file in the interface, refer to its name and check that the tool can read it: mentioning a file in a prompt does not upload it.

Output format

Specify audience, language, length limit and fields. A blank table or a brief reference answer can make the expected format clearer.

Editorial team: Take Two. Edition 7 October 2026.

One request, four blocks

```
<request>
  <goal>
    [Specific result, audience and context]
  </goal>
  <method>
    [Useful steps and verification criteria]
    Use attachments as material to analyse.
    Flag missing information before concluding.
    Separate facts, assumptions and suggestions.
  </method>
  <attachments>
    <document name="source-1">
      [Relevant text, anonymised when needed]
    </document>
  </attachments>
  <output_format>
    [Structure, language, length and required fields]
  </output_format>
</request>
```

These tags are a practical convention, not a mandatory standard or a defence against instructions contained in documents. Results also depend on the model, inputs and checks. If text includes <, > or &, use <, > and & or a CDATA section in an XML file.

A task to try today

Pick a recurring task, use fictional or anonymised data and fill in the four blocks. Try the prompt, check the answer and note one useful change. Keep the version that works best for that task.

Go further: the first course covers the basics; the second develops XML-like prompting; the third helps compare tools and plans.

03 / VERIFICATION

Before using the answer

A well-written answer may contain errors. For work tasks, check these seven points and refine the prompt using the problems you find.

01 Goal

Does the answer complete the requested task for the right audience?

02 Facts and numbers

Compare names, dates, quantities and calculations with the source material. Recalculate at least the steps that affect the decision.

03 Sources

Open cited sources and verify that they exist, are current and actually support the claim. A plausible citation is not enough.

04 Missing information

Can you distinguish supplied facts, assumptions and points to clarify? Avoid treating absent information as certainty.

05 Format

Do length, language, tone and fields match the request? Also check for omitted rows and duplicate entries.

06 Confidentiality

Does the result include personal or business information that should not be shared? Use only material you are authorised to process in your chosen tool.

07 Human review

Before sending, publishing or executing a procedure, a competent person should verify the relevant parts and own the decision.

If an answer is weak, change one element at a time: supply a missing fact, clarify the audience or show the desired format. Judge the revised answer using the same criteria.

EXAMPLE 1 / 3

Write an email to a client

Before

Write a professional email about a delay.

After

```
<request>
  <goal>Tell the client that delivery moves from 12 to 15 November. Ask
    them to confirm the new date.</goal>
  <method>Use only supplied facts. Accept responsibility without inventing
    causes or promises. Keep the tone direct and polite.</method>
  <attachments>Project: portal upgrade. Verified reason: testing is still
    underway. Contact: Marta, a fictional name.</attachments>
  <output_format>Subject and body in English, up to 120 words. End with
    one request to confirm.</output_format>
</request>
```

Why this helps

The first version leaves dates, audience and reasons to the model. The second defines the facts, responsibility and action requested from the client.

What to check

Check for both dates, testing and the confirmation request. There should be no discounts, invented technical causes or unagreed guarantees.

These are original teaching examples with fictional data. The expected outcomes describe how to judge results; they are not measured or recorded model responses.

EXAMPLE 2 / 3

Turn meeting notes into tasks

Before

Summarise this meeting.

After

```
<request>
  <goal>Turn notes into a task list to share with the team.</goal>
  <method>Separate decisions and tasks. Do not invent owners or deadlines.
    Write "to be agreed" in missing fields.</method>
  <attachments>Luca checks the backup by Friday. The team chooses a two-
    stage release. The manual needs updating; no owner or deadline agreed.
    Fictional names and scenario.</attachments>
  <output_format>One sentence on decisions, then a table with task, owner,
    deadline and open question.</output_format>
</request>
```

Why this helps

A summary can lose the operational actions. The requested format shows what the team needs to work on and what still needs agreement.

What to check

The backup should name Luca and Friday; the manual should have "to be agreed" for both owner and deadline. The two-stage release is a decision, not a deadline.

These are original teaching examples with fictional data. The expected outcomes describe how to judge results; they are not measured or recorded model responses.

EXAMPLE 3 / 3

Compare two proposals

Before

Which quote is better?

After

```
<request>
  <goal>Prepare a preliminary comparison of two proposals for an
    information website. The available budget is EUR 5,000.</goal>
  <method>Compare only supplied facts. Do not assume an unmentioned
    service is included or excluded. Flag questions for suppliers before
    deciding.</method>
  <attachments>A: EUR 4,200, 6 weeks, maintenance unspecified. B: EUR
    4,800, 8 weeks, 12 months of maintenance included. VAT and content
    unspecified in both. Fictional proposals.</attachments>
  <output_format>Table with cost, schedule, maintenance and unknowns; then
    three supplier questions. No final choice based on this data
    alone.</output_format>
</request>
```

Why this helps

“Better” depends on criteria. The revised prompt states the budget and avoids treating unknown conditions as confirmed facts.

What to check

Both prices are below EUR 5,000 before any VAT; the total budget still needs checking. Questions should cover VAT, content and maintenance in A.

These are original teaching examples with fictional data. The expected outcomes describe how to judge results; they are not measured or recorded model responses.

Approfondimento / Further reading: Anthropic, “Structure prompts with XML tags”, consultato / checked 7 October 2026.